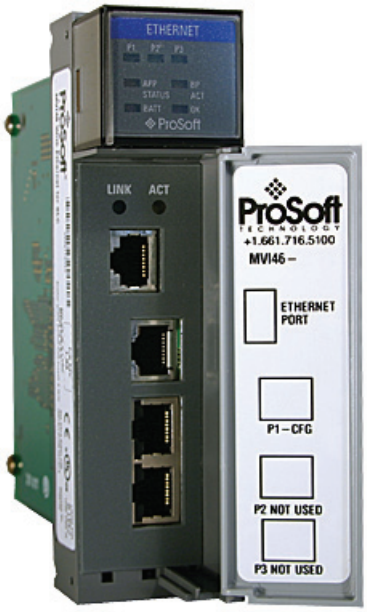
ProSoft TECHNOLOGY

Where Automation Connects.



inRAx®

MVI46-MNETC

SLC Platform

Modbus TCP/IP Communication Client Module

August 27, 2009

USER MANUAL

Important Installation Instructions

Power, Input and Output (I/O) wiring must be in accordance with Class I, Division 2 wiring methods, Article 501-4 (b) of the National Electrical Code, NFPA 70 for installation in the U.S., or as specified in Section 18-1J2 of the Canadian Electrical Code for installations in Canada, and in accordance with the authority having jurisdiction. The following warnings must be heeded:

A WARNING - EXPLOSION HAZARD - SUBSTITUTION OF COMPONENTS MAY IMPAIR SUITABILITY FOR CLASS I, DIV. 2;
B WARNING - EXPLOSION HAZARD - WHEN IN HAZARDOUS LOCATIONS, TURN OFF POWER BEFORE REPLACING OR WIRING MODULES
C WARNING - EXPLOSION HAZARD - DO NOT DISCONNECT EQUIPMENT UNLESS POWER HAS BEEN SWITCHED OFF OR THE AREA IS KNOWN TO BE NONHAZARDOUS.
D THIS DEVICE SHALL BE POWERED BY CLASS 2 OUTPUTS ONLY.

MVI (Multi Vendor Interface) Modules

WARNING - EXPLOSION HAZARD - DO NOT DISCONNECT EQUIPMENT UNLESS POWER HAS BEEN SWITCHED OFF OR THE AREA IS KNOWN TO BE NON-HAZARDOUS.

AVERTISSEMENT - RISQUE D'EXPLOSION - AVANT DE DÉCONNECTER L'EQUIPMENT, COUPER LE COURANT OU S'ASSURER QUE L'EMPLACEMENT EST DÉSIGNÉ NON DANGEREUX.

CL I Div 2 GPs A, B, C, D

Temp Code T5

II 3 G

Ex nA IIC T5 X

0° C <= Ta <= 60° C

II - Equipment intended for above ground use (not for use in mines).

3 - Category 3 equipment, investigated for normal operation only.

G - Equipment protected against explosive gasses.

Warnings

North America Warnings

A Warning - Explosion Hazard - Substitution of components may impair suitability for Class I, Division 2.
B Warning - Explosion Hazard - When in Hazardous Locations, turn off power before replacing or rewiring modules.
Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be nonhazardous.
C Suitable for use in Class I, division 2 Groups A, B, C and D Hazardous Locations or Non-Hazardous Locations.

ATEX Warnings and Conditions of Safe Usage:

Power, Input, and Output (I/O) wiring must be in accordance with the authority having jurisdiction

A Warning - Explosion Hazard - When in hazardous locations, turn off power before replacing or wiring modules.
B Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous.
C These products are intended to be mounted in an IP54 enclosure. The devices shall provide external means to prevent the rated voltage being exceeded by transient disturbances of more than 40%. This device must be used only with ATEX certified backplanes.
D DO NOT OPEN WHEN ENERGIZED.

Electrical Ratings

- Backplane Current Load: 800 mA @ 5 V DC; 3mA @ 24V DC
- Operating Temperature: 0 to 60°C (32 to 140°F)
- Storage Temperature: -40 to 85°C (-40 to 185°F)
- Shock: 30g Operational; 50g non-operational; Vibration: 5 g from 10 to 150 Hz
- Relative Humidity 5% to 95% (non-condensing)
- All phase conductor sizes must be at least 1.3 mm(squared) and all earth ground conductors must be at least 4mm(squared).

Markings

ANSI / ISA	ISA 12.12.01 Class I Division 2, GPs A, B, C, D
CSA/cUL	C22.2 No. 213-1987
CSA CB Certified	IEC61010
ATEX	EN60079-0 Category 3, Zone 2 EN60079-15

243333

Warning: This module is not hot-swappable! Always remove power from the rack before inserting or removing this module, or damage may result to the module, the processor, or other connected devices.

Battery Life Advisory

MVI46, MVI56, MVI69 and MVI71 series use a rechargeable Lithium Vanadium Pentoxide battery to backup the 512K SRAM memory, real-time clock, and CMOS. The battery should last for the life of the module.

The module must be powered for approximately twenty hours before it becomes fully charged. After it is fully charged, the battery provides backup power for the CMOS setup and configuration data, the real-time clock, and the 512K SRAM memory for approximately 21 days.

Before you remove a module from its power source, ensure that the battery within the module is fully charged. A fully charged battery will hold the BIOS settings (after being removed from its power source) for a limited number of days. When the battery is fully discharged, the module will revert to the default BIOS settings.

Note: The battery is not user replaceable.

Your Feedback Please

We always want you to feel that you made the right decision to use our products. If you have suggestions, comments, compliments or complaints about the product, documentation, or support, please write or call us.

ProSoft Technology
5201 Truxtun Ave., 3rd Floor
Bakersfield, CA 93309
+1 (661) 716-5100
+1 (661) 716-5101 (Fax)
www.prosoft-technology.com
support@prosoft-technology.com

Copyright © ProSoft Technology, Inc. 2009. All Rights Reserved.

MVI46-MNETC User Manual
August 27, 2009

ProSoft Technology ®, ProLinx ®, inRAx ®, ProTalk®, and RadioLinx ® are Registered Trademarks of ProSoft Technology, Inc. All other brand or product names are or may be trademarks of, and are used to identify products and services of, their respective owners.

ProSoft Technology® Product Documentation

In an effort to conserve paper, ProSoft Technology no longer includes printed manuals with our product shipments. User Manuals, Datasheets, Sample Ladder Files, and Configuration Files are provided on the enclosed CD-ROM, and are available at no charge from our web site: www.prosoft-technology.com

Printed documentation is available for purchase. Contact ProSoft Technology for pricing and availability.

North America: +1.661.716.5100

Asia Pacific: +603.7724.2080

Europe, Middle East, Africa: +33 (0) 5.3436.87.20

Latin America: +1.281.298.9109

Contents

Important Installation Instructions 2
MVI (Multi Vendor Interface) Modules 2
Warnings 2
Battery Life Advisory 3
Your Feedback Please 4
ProSoft Technology® Product Documentation 4

Guide to the MVI46-MNETC User Manual 7

1 Start Here 9

1.1 System Requirements - MVI46 PCB 9
1.2 Package Contents 10
1.3 Install ProSoft Configuration Builder Software 10
1.4 Setting Jumpers 11
1.5 Install the Module in the Rack 12
1.6 Connect your PC to the Processor 13
1.7 Download the Sample Program to the Processor 14
1.8 Connect your PC to the Module 17

2 Configuring the MVI46-MNETC Module 19

2.1 Using ProSoft Configuration Builder 19
2.2 Setting Command Control Bits 34
2.3 Configuring the Floating Point Data Transfer 34
2.4 Download the Project to the Module 39

3 Ladder Logic 41

3.1 Adding the Module to an Existing Project 41

4 Diagnostics and Troubleshooting 45

4.1 Reading Status Data from the Module 45
4.2 LED Status Indicators 56

5 Reference 59

5.1 Product Specifications 59
5.2 Functional Overview 61
5.3 Cable Connections 73

6 Support, Service & Warranty 77

6.1 How to Contact Us: Technical Support 77
6.2 Return Material Authorization (RMA) Policies and Conditions 78
6.3 LIMITED WARRANTY 79

Index 83

Guide to the MVI46-MNETC User Manual

Function		Section to Read	Details
Introduction (Must Do)	→	Start Here (page 9)	This Section introduces the customer to the module. Included are: package contents, system requirements, hardware installation, and basic configuration.
Verify Communication, Diagnostic and Troubleshooting	→	Verifying Communication (page 56) Diagnostics and Troubleshooting (page 45)	This section describes how to verify communications with the network. Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Overview	→	Reference (page 59) Functional Overview (page 61) Product Specifications (page 59)	These sections contain general references associated with this product, Specifications, and the Functional Overview.
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 77)	This section contains Support, Service and Warranty information. Index of chapters.

1 Start Here

In This Chapter

- System Requirements - MVI46 PCB 9
- Package Contents 10
- Install ProSoft Configuration Builder Software 10
- Setting Jumpers 11
- Install the Module in the Rack 12
- Connect your PC to the Processor 13
- Download the Sample Program to the Processor 14
- Connect your PC to the Module 17

To get the most benefit from this User Manual, you should have the following skills:

- **Rockwell Automation® RSLogix™ software**: launch the program, configure ladder logic, and transfer the ladder logic to the processor
- **Microsoft Windows:** install and launch programs, execute menu commands, navigate dialog boxes, and enter data.
- **Hardware installation and wiring**: install the module, and safely connect MNETC and SLC devices to a power source and to the MVI46-MNETC module's application port(s).

Caution: You must be able to complete the application without exposing personnel or equipment to unsafe or inappropriate working conditions.

1.1 System Requirements - MVI46 PCB

The MVI46-MNETC module requires the following minimum hardware and software components:

- Rockwell Automation SLC 5/02 M0/M1 capable processors (or newer), with compatible power supply and one free slot in the rack, for the MVI46-MNETC module. The module requires 800mA of available power.
- Rockwell Automation RSLogix 500 programming software.
- Rockwell Automation RSLinx communication software
- Pentium® II 500 MHz minimum. Pentium III 733 MHz (or better) recommended
- Supported operating systems:
  - Microsoft® Windows 98
  - Windows NT® (version 4 with SP4 or higher)

- Windows 2000
- Windows XP

- 32 Mbytes of RAM minimum, 64 Mbytes of RAM recommended
- 50 Mbytes of free hard disk space (or more based on application requirements)
- 16-color VGA graphics adapter, 640 x 480 minimum resolution (256 Color 800 × 600 recommended)
- CD-ROM drive

1.2 Package Contents

The following components are included with your MVI46-MNETC module, and are all required for installation and configuration.

Important: Before beginning the installation, please verify that all of the following items are present.

Qty.	Part Name	Part Number	Part Description
1	MVI46-MNETC Module	MVI46-MNETC	Modbus TCP/IP Communication Client Module
1	Cable	Cable #15, RS232 Null Modem	For RS232 Connection to the CFG Port
1	Cable	RJ45 to DB9 Male Adapter	For DB9 Connection to Module's Port
1	inRAx Solutions CD		Contains sample programs, utilities and documentation for the MVI46-MNETC module.

If any of these components are missing, please contact ProSoft Technology Support for replacement parts.

1.3 Install ProSoft Configuration Builder Software

You must install the ProSoft Configuration Builder (PCB) software in order to configure the module. You can always get the newest version of ProSoft Configuration Builder from the ProSoft Technology web site.

To install ProSoft Configuration Builder from the ProSoft Web Site

1. Open your web browser and navigate to *http://www.prosoft-technology.com/pcb*
2. Click the **DOWNLOAD HERE** link to download the latest version of ProSoft Configuration Builder.
3. Choose **"SAVE"** or **"SAVE FILE"** when prompted.
4. Save the file to your Windows Desktop, so that you can find it easily when you have finished downloading.
5. When the download is complete, locate and open the file, and then follow the instructions on your screen to install the program.

If you do not have access to the Internet, you can install ProSoft Configuration Builder from the ProSoft Solutions CD-ROM, included in the package with your module.

To install ProSoft Configuration Builder from the Product CD-ROM

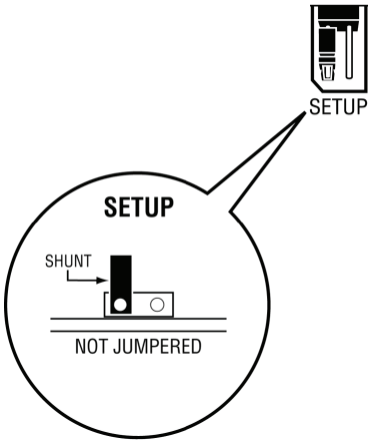
1. Insert the ProSoft Solutions Product CD-ROM into the CD-ROM drive of your PC. Wait for the startup screen to appear.
2. On the startup screen, click **PRODUCT DOCUMENTATION**. This action opens an Windows Explorer file tree window.
3. Click to open the **UTILITIES** folder. This folder contains all of the applications and files you will need to set up and configure your module.
4. Double-click the **SETUPCONFIGURATIONTOOL** folder, double-click the **"PCB_*.EXE"** file and follow the instructions on your screen to install the software on your PC. The information represented by the **"*"** character in the file name is the PCB version number and, therefore, subject to change as new versions of PCB are released.

Note: Many of the configuration and maintenance procedures use files and other utilities on the CD-ROM. You may wish to copy the files from the Utilities folder on the CD-ROM to a convenient location on your hard drive.

1.4 Setting Jumpers

The Setup Jumper acts as "write protection" for the module's flash memory. In "write protected" mode, the Setup pins are not connected, and the module's firmware cannot be overwritten. Do not jumper the Setup pins together unless you are directed to do so by ProSoft Technical Support.

The following illustration shows the MVI46-MNETC jumper configuration.



Note: If you are installing the module in a remote rack, you may prefer to leave the Setup pins jumpered. That way, you can update the module's firmware without requiring physical access to the module.

1.5 Install the Module in the Rack

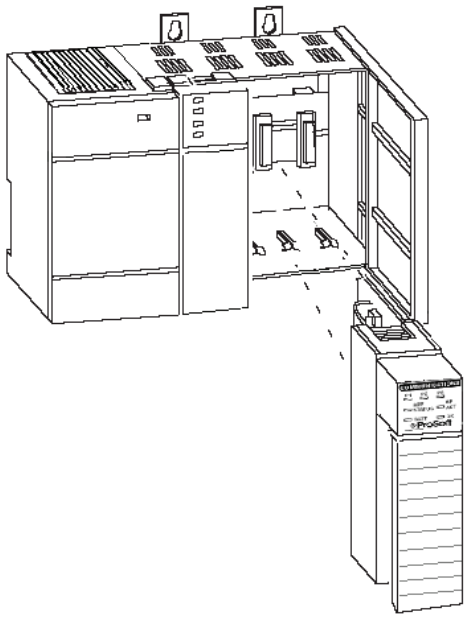
If you have not already installed and configured your SLC processor and power supply, please do so before installing the MVI46-MNETC module. Refer to your Rockwell Automation product documentation for installation instructions.

Warning: You must follow all safety instructions when installing this or any other electronic devices. Failure to follow safety procedures could result in damage to hardware or data, or even serious injury or death to personnel. Refer to the documentation for each device you plan to connect to verify that suitable safety procedures are in place before installing or servicing the device.

After you have checked the placement of the jumpers, insert MVI46-MNETC into the SLC™ chassis. Use the same technique recommended by Rockwell Automation to remove and install SLC™ modules.

Warning: This module is not hot-swappable! Always remove power from the rack before inserting or removing this module, or damage may result to the module, the processor, or other connected devices.

1. Turn power OFF.
2. Align the module with the top and bottom guides, and slide it into the rack until the module is firmly against the backplane connector.



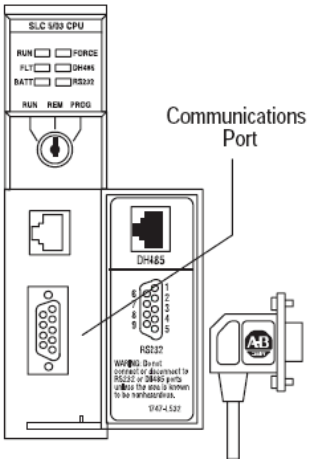
3. With a firm but steady push, snap the module into place.
4. Check that the holding clips on the top and bottom of the module are securely in the locking holes of the rack.

5 Make a note of the slot location. You will need to identify the slot in which the module is installed in order for the sample program to work correctly. Slot numbers are identified on the green circuit board (backplane) of the SLC rack.
6 Turn power ON.

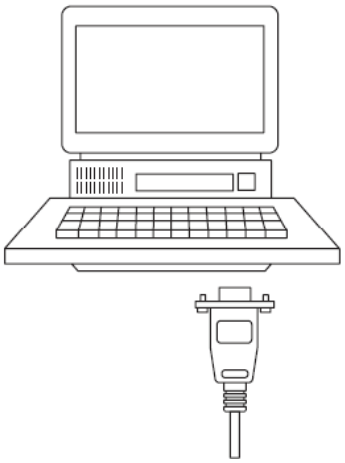
Note: If you insert the module improperly, the system may stop working, or may behave unpredictably.

1.6 Connect your PC to the Processor

1 Connect the right-angle connector end of the cable to your controller at the communications port.



2 Connect the straight connector end of the cable to the serial port on your computer.

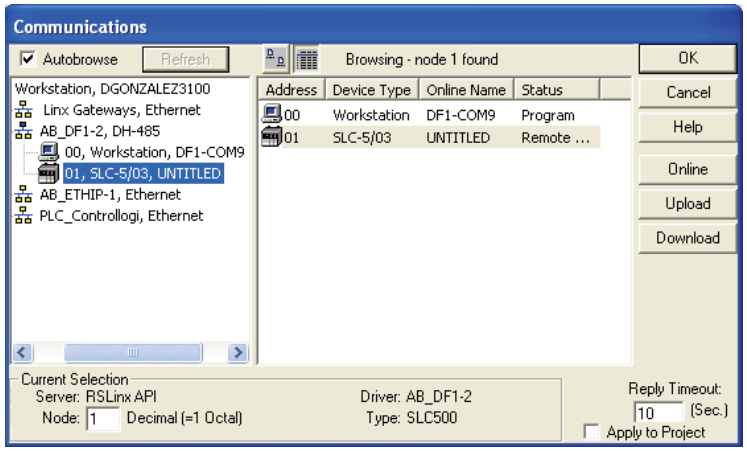


1.7 Download the Sample Program to the Processor

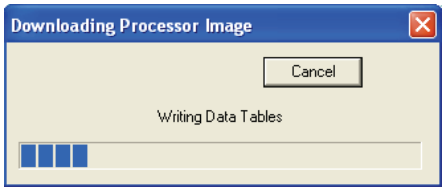
To download the sample program from RSLogix 500 to the SLC processor:

Note: The key switch on the front of the SLC processor must be in the REM position.

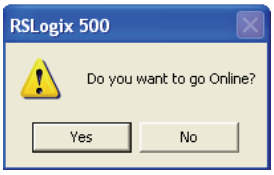
1 If you are not already online to the processor, open the Communications menu, and then choose Download. RSLogix will establish communication with the processor.



2 Click the Download button to transfer the sample program to the processor.
3 RSLogix will compile the program and transfer it to the processor. This process may take a few minutes.



4 When the download is complete, RSLogix will open another confirmation dialog box. Click Yes to switch the processor from Program mode to Run mode.

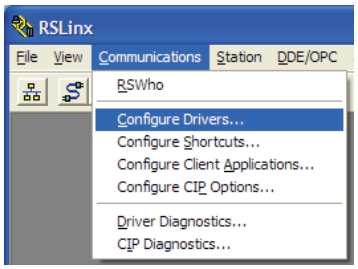


Note: If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

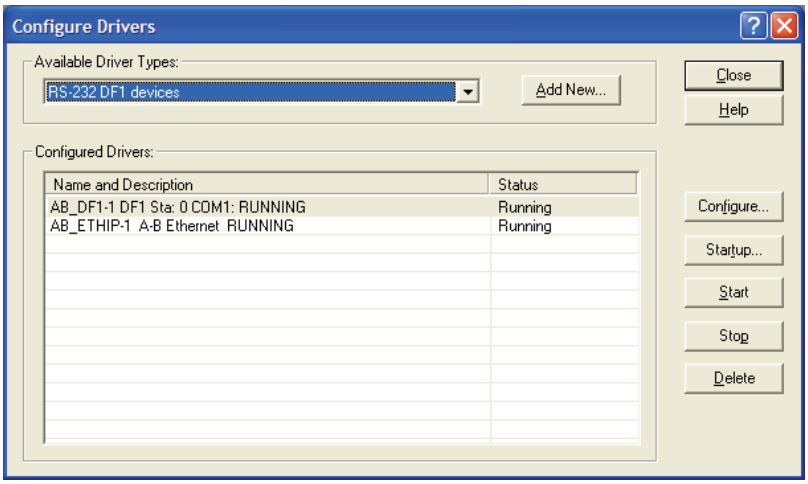
1.7.1 Configuring the RSLinx Driver for the PC COM Port

If RSLogix is unable to establish communication with the processor, follow these steps

1 Open **RSLINX.**
2 Open the **COMMUNICATIONS** menu, and choose **CONFIGURE DRIVERS**.



This action opens the **CONFIGURE DRIVERS** dialog box.



Note: If the list of configured drivers is blank, you must first choose and configure a driver from the Available Driver Types list. The recommended driver type to choose for serial communication with the processor is **RS-232 DF1 DEVICES**.

3 Click to select the driver, and then click **CONFIGURE.** This action opens the **CONFIGURE ALLEN-BRADLEY DF1 COMMUNICATIONS DEVICE** dialog box.

4 Click the **AUTO-CONFIGURE** button. RSLinx will attempt to configure your serial port to work with the selected driver.

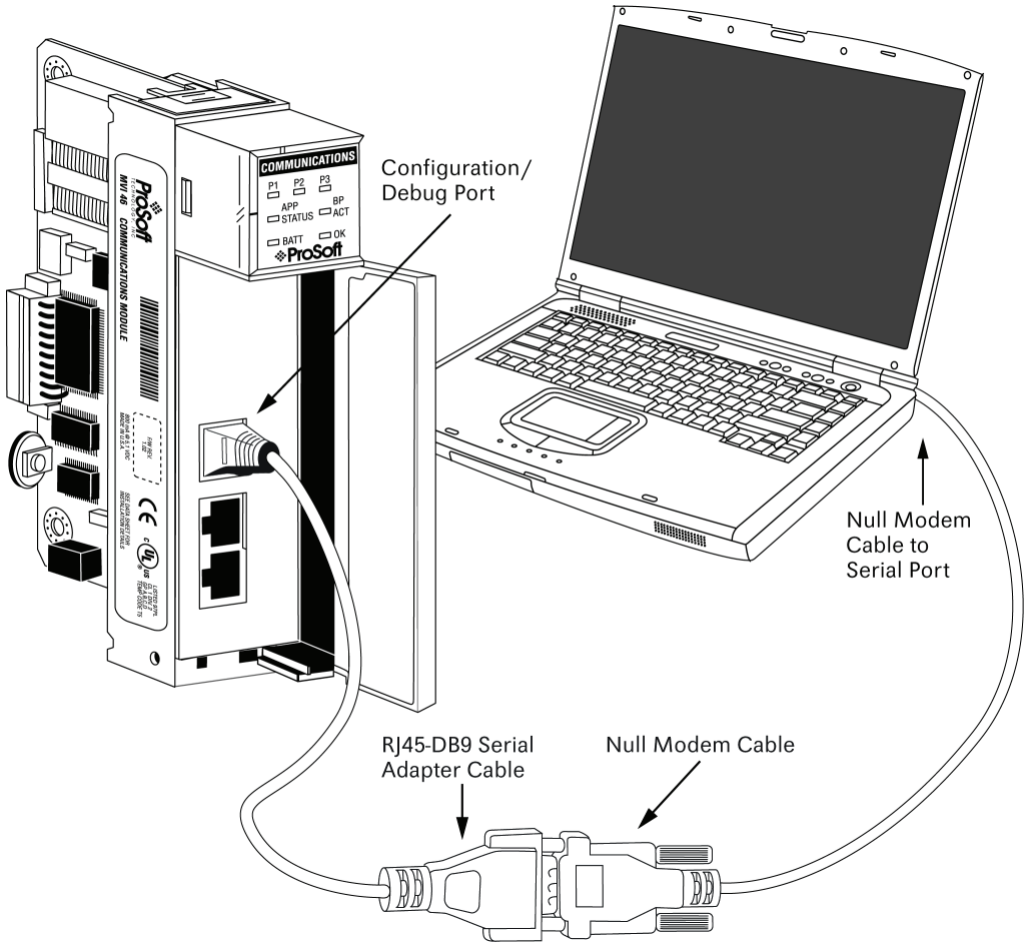
5 When you see the message **AUTO CONFIGURATION SUCCESSFUL**, click the **OK** button to dismiss the dialog box.

Note: If the auto-configuration procedure fails, verify that the cables are connected correctly between the processor and the serial port on your computer, and then try again. If you are still unable to auto-configure the port, refer to your RSLinx documentation for further troubleshooting steps.

1.8 Connect your PC to the Module

With the module securely mounted, connect your PC to the Configuration/Debug port using an RJ45-DB-9 Serial Adapter Cable and a Null Modem Cable.

1. Attach both cables as shown.
2. Insert the RJ45 cable connector into the Configuration/Debug port of the module.
3. Attach the other end to the serial port on your PC.



2 Configuring the MVI46-MNETC Module

In This Chapter

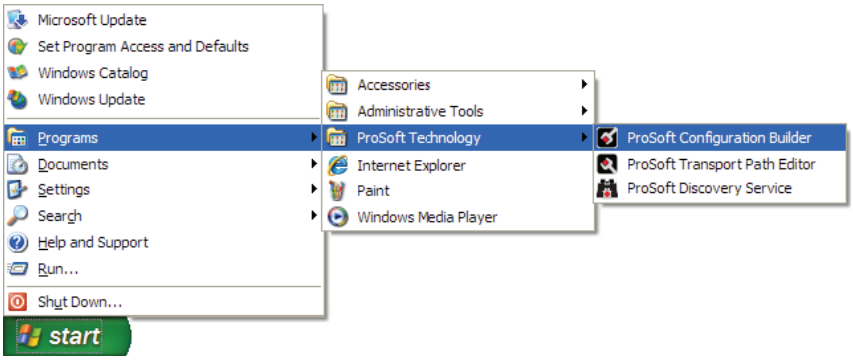
- Using ProSoft Configuration Builder 19
- Setting Command Control Bits 34
- Configuring the Floating Point Data Transfer 34
- Download the Project to the Module 39

2.1 Using ProSoft Configuration Builder

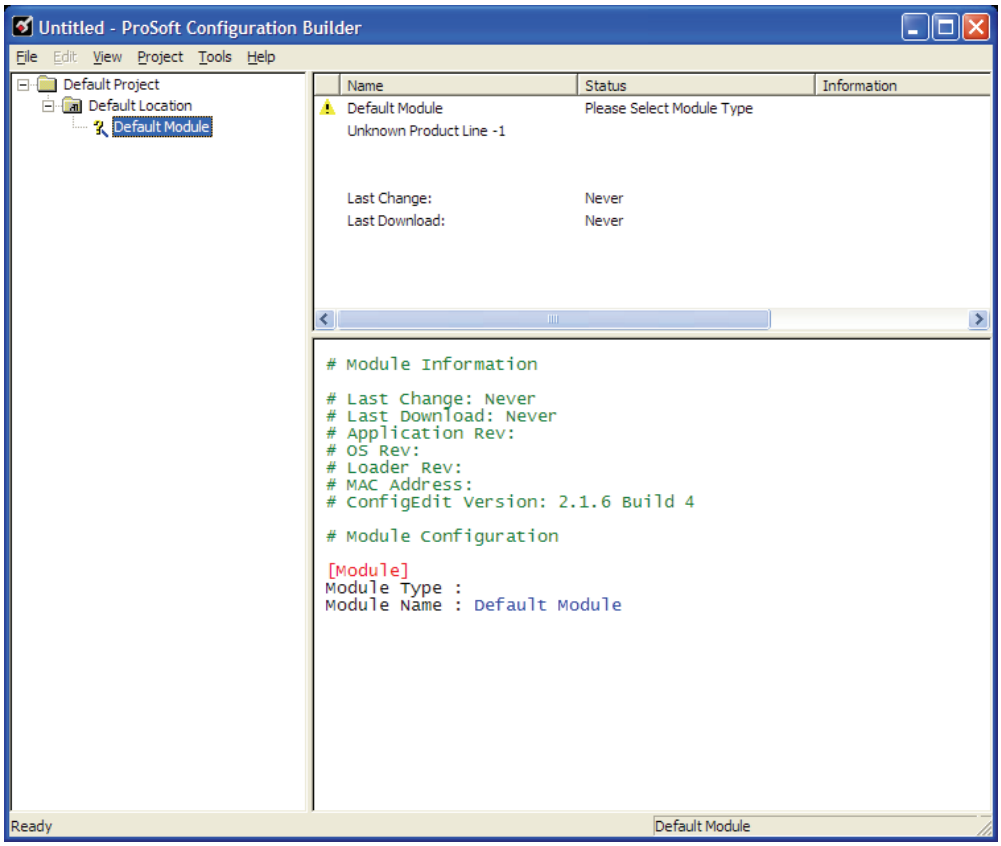
ProSoft Configuration Builder (PCB) provides a quick and easy way to manage module configuration files customized to meet your application needs. *PCB* is not only a powerful solution for new configuration files, but also allows you to import information from previously installed (known working) configurations to new projects.

2.1.1 Set Up the Project

To begin, start **PROSOFT CONFIGURATION BUILDER** (PCB).

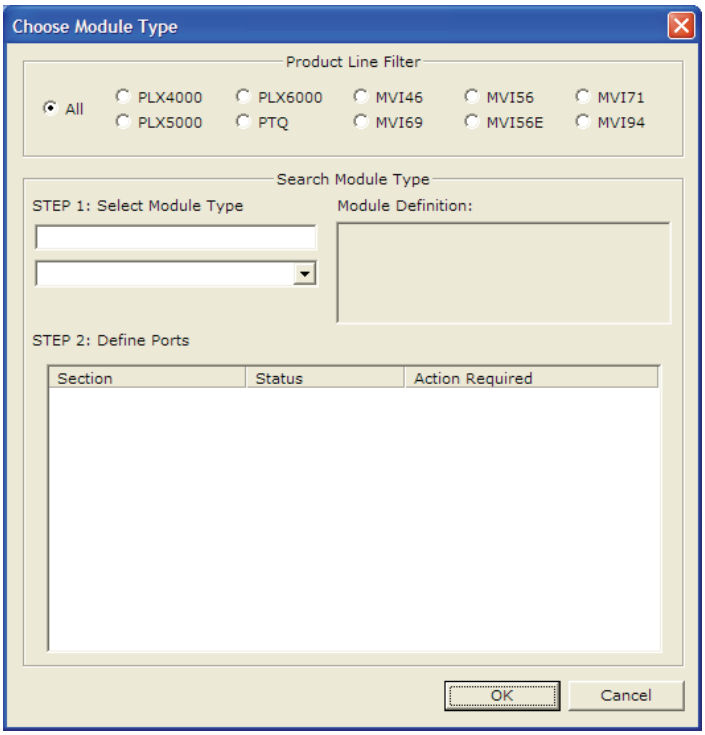


If you have used other Windows configuration tools before, you will find the screen layout familiar. PCB's window consists of a tree view on the left, and an information pane and a configuration pane on the right side of the window. When you first start *PCB*, the tree view consists of folders for **DEFAULT PROJECT** and **DEFAULT LOCATION**, with a **DEFAULT MODULE** in the Default Location folder. The following illustration shows the *PCB* window with a new project.



Your first task is to add the MVI46-MNETC module to the project.

1. Use the mouse to select "Default Module" in the tree view, and then click the right mouse button to open a shortcut menu.
2. On the shortcut menu, choose "Choose Module Type". This action opens the Choose Module Type dialog box.

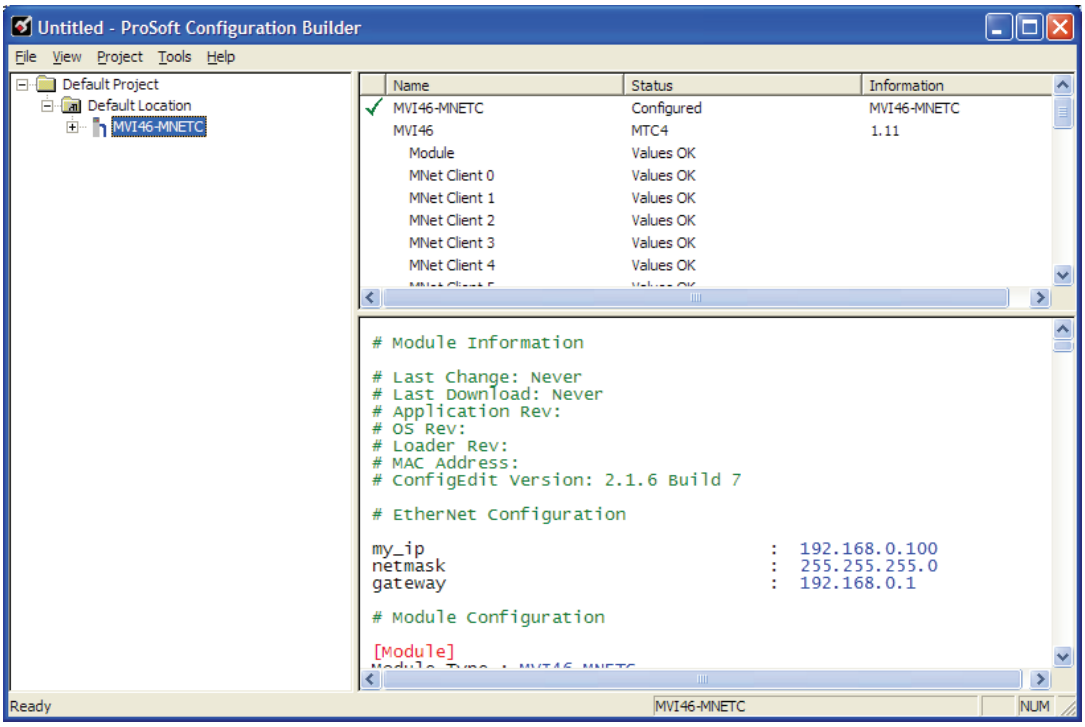


3. In the **PRODUCT LINE FILTER** area of the dialog box, select MVI46.
4. In the **SELECT MODULE TYPE** dropdown list, select MVI46-MNETC, and then click **OK** to save your settings and return to the ProSoft Configuration Builder window.

The next task is to set the module parameters.

2.1.2 Set Module Parameters

The next task is to configure module parameters. Notice that the contents of the information pane and the configuration pane changed when you added the MVI46-MNETC module to the project.



At this time, you may wish to rename the "Default Project" and "Default Location" folders in the tree view.

To rename an object:

1. Select the object, and then click the right mouse button to open a shortcut menu. From the shortcut menu, choose **RENAME.**
2. Type the name to assign to the object.
3. Click away from the object to save the new name.

Configuring Module Parameters

1. Click on the plus sign next to the icon to expand module information.
2. Double-click the icon to open the **EDIT** dialog box.
3. To edit a parameter, select the parameter in the left pane and make your changes in the right pane.
4. Click **OK** to save your changes.

Printing a Configuration File

1. Select the **MODULE** icon, and then click the right mouse button to open a shortcut menu.
2. On the shortcut menu, choose **VIEW CONFIGURATION.** This action opens the **VIEW CONFIGURATION** window.
3. On the **VIEW CONFIGURATION** window, open the **FILE** menu, and choose **PRINT.** This action opens the **PRINT** dialog box.
4. On the **PRINT** dialog box, choose the printer to use from the dropdown list, select printing options, and then click **OK.**

2.1.3 [Module]

This section of the configuration describes the database setup and module level parameters. This section provides the module with a unique name, identifies the method of failure for the communications for the module if the processor is not in run, and describes how to initialize the module upon startup.

Error/Status Pointer

-1 to 4955

Starting register location in virtual Modbus database for the error/status table. If a value of -1 is entered, the error/status data will not be placed in the database. All other valid values determine the starting location of the data. This data area includes the module version information and all error/status data.

M1 Write Size

0 to 4000

This parameter limits the M1 data transferred from the processor to the module. The module application automatically adjusts the size to an even 50-word boundary as this is the minimum data transfer size for the application. For example, a value of 199 would automatically be adjusted to 200. This feature improves the transfer of data from the processor to the module.

Failure Flag Count

0 through 65535

This parameter specifies the number of successive transfer errors that must occur before halting communication on the application port(s). If the parameter is set to 0, the application port(s) will continue to operate under all conditions. If the value is set larger than 0 (1 to 65535), communications will cease if the specified number of failures occur.

Initialize Output Data

0 = No

1 = Yes

This parameter is used to determine if the output data for the module should be initialized with values from the processor. If the value is set to 0, the output data will be initialized to 0. If the value is set to 1, the data will be initialized with data from the processor. Use of this option requires associated ladder logic to pass the data from the processor to the module.

Duplex/Speed Code

0, 1, 2, 3 or 4

This parameter allows you to force the module to use a specific duplex and speed setting.

- Value = 1: Half duplex, 10 MB speed
- Value = 2: Full duplex, 10 MB speed
- Value = 3: Half duplex, 100 MB speed
- Value = 4: Full duplex, 100 MB speed
- Value = 0: Auto negotiate.

Auto Negotiate is the default value for backward compatibility. This feature is not implemented in older software revisions.

2.1.4 [Static ARP Table]

The Static ARP Table defines a list of static IP addresses that the module will use when an ARP (Address Resolution Protocol) is required. The module will accept up to 40 static IP/MAC address data sets.

Use the Static ARP table to reduce the amount of network traffic by specifying IP addresses and their associated MAC (hardware) addresses that the MVI46-MNETC module will be communicating with regularly.

Important: If the device in the field is changed, this table must be updated to contain the new MAC address for the device and downloaded to the module. If the MAC is not changed, no communications with the module will be provided.

IP Address

Dotted notation

This table contains a list of static IP addresses that the module will use when an ARP is required. The module will accept up to 40 static IP/MAC address data sets.

Important: If the device in the field is changed, this table must be updated to contain the new MAC address for the device and downloaded to the module. If the MAC is not changed, no communications with the module will occur.

Hardware MAC Address

Hex Value

This table contains a list of static MAC addresses that the module will use when an ARP is required. The module will accept up to 40 static IP/MAC address data sets.

Important: If the device in the field is changed, this table must be updated to contain the new MAC address for the device and downloaded to the module. If the MAC is not changed, no communications with the module will occur.

2.1.5 [MNET Client x]

This section defines general configuration for the MNET Client (Master).

Error/Status Pointer

-1 to 4990

Starting register location in virtual database for the error/status table for this client. If a value of -1 is entered, the error/status data will not be placed in the database. All other valid values determine the starting location of the data.

Command Error Pointer

-1 to 4999

This parameter sets the address in the internal database where the command error data will be placed. If the value is set to -1, the data will not be transferred to the database.

Minimum Command Delay

0 to 65535

This parameter specifies the number of milliseconds to wait between the initial issuance of a command. This parameter can be used to delay all commands sent to slaves to avoid "flooding" commands on the network. This parameter does not affect retries of a command as they will be issued when failure is recognized.

Response Timeout

0 to 65535 milliseconds

This is the time in milliseconds that a client will wait before re-transmitting a command if no response is received from the addressed server. The value to use depends upon the type of communication network used, and the expected response time of the slowest device on the network.

Retry Count

0 to 10

This parameter specifies the number of times a command will be retried if it fails.

Float Flag

Yes or No

This flag specifies how the Slave driver will respond to Function Code 3, 6, and 16 commands (read and write Holding Registers) from a remote Master when it is moving 32-bit floating-point data.

If the remote Master expects to receive or will send one, complete, 32-bit floating-point value for each count of one (1), then set this parameter to **YES**, especially if the Master must read or write from Modbus addresses above gateway address 3999 (virtual Modbus address 44000 or 440000). When set to **YES**, the Slave driver will return values from two, consecutive, 16-bit internal memory registers (32 total bits) for each count in the read command or receive 32-bits per count from the Master for write commands. Example: Count = 10, Slave driver will send 20 16-bit registers for 10 total 32-bit floating-point values.

If, however, the remote Master sends a count of two (2) for each 32-bit floating-point value it expects to receive or send, or, if you do not plan to use floating-point data in your application, then set this parameter to **NO,** which is the default setting.

You will also need to set the *Float Start* and *Float Offset* parameters to appropriate values whenever the *Float Flag* parameter is set to **YES**.

Float Start

0 to 65535

This parameter defines the first register of floating-point data. All requests with register values greater-than or equal to this value will be considered floating-point data requests. This parameter is only used if the Float Flag is enabled. For example, if a value of 7000 is entered, all requests for registers 7000 and above will be considered as floating-point data.

Float Offset

0 to 9999

This parameter defines the start register for floating-point data in the internal database. This parameter is used only if the Float Flag is enabled. For example, if the Float Offset value is set to 3000 and the float start parameter is set to 7000, data requests for register 7000 will use the internal Modbus register 3000.

ARP Timeout

1 to 60

This parameter specifies the number of seconds to wait for an ARP reply after a request is issued.

Command Error Delay

0 to 300

This parameter specifies the number of 100 millisecond intervals to turn off a command in the error list after an error is recognized for the command. If this parameter is set to 0, there will be no delay.

MBAP Port Override

Yes or No

If this parameter is set to Y (Yes), all messages generated by the client driver will be MBAP format messages to all Service Port values.

If this parameter is set to N (No) (default value), or is omitted from the configuration file, all messages sent to Service Port 502 will be MBAP format messages, and all other Service Ports values will use the encapsulated Modbus message format (MNET).

Each client is configured independently in the configuration file.

This parameter applies to firmware version 1.05 and above. For downward compatibility, you may omit this parameter from the client's configuration.

2.1.6 [MNET Client x Commands]

The **[MNET CLIENT X COMMANDS]** section of the configuration sets the Modbus TCP/IP Client command list. This command list polls Modbus TCP/IP Server devices attached to the Modbus TCP/IP Client port. The module supports numerous commands. This permits the module to interface with a wide variety of Modbus TCP/IP protocol devices.

The function codes used for each command are those specified in the Modbus protocol. Each command list record has the same format. The first part of the record contains the information relating to the MVI46-MNETC communication module, and the second part contains information required to interface to the Modbus TCP/IP Server device.

Command List Overview

In order to interface the ProLinx module with Modbus TCP/IP Server devices, you must construct a command list. The commands in the list specify the Server device to be addressed, the function to be performed (read or write), the data area in the device to interface with and the registers in the internal database to be associated with the device data. The Client command list supports up to 16 commands.

The command list is processed from top (command #0) to bottom. A poll interval parameter is associated with each command to specify a minimum delay time in tenths of a second between the issuance of a command. If the user specifies a value of 10 for the parameter, the command will be executed no more frequently than every 1 second.

Commands Supported by the Module

The format of each command in the list depends on the Modbus Function Code being executed. The following table describes the functions supported by the module:

Function Code	Definition	Supported in Client
1	Read Coil Status	X
2	Read Input Status	X
3	Read Holding Registers	X
4	Read Input Registers	X
5	Force (Write) Single Coil	X
6	Preset (Write) Single Register	X
15	Force (Write) Multiple Coils	X
16	Preset (Write) Multiple Registers	X

Each command list record has the same general format. The first part of the record contains the information relating to the communication module and the second part contains information required to interface to the Modbus TCP/IP Server device.

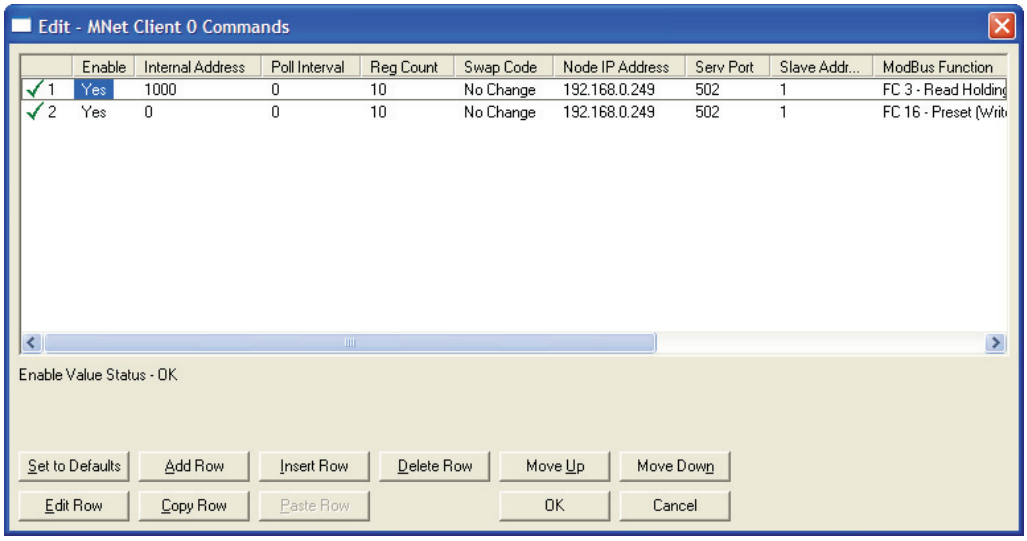
Command Entry Formats

The following table shows the structure of the configuration data necessary for each of the supported commands.

Column #	1	2	3	4	5	6	7	8	9	10
Function Code	Enable Code	Internal Address	Poll Interval Time	Count	Swap Code	IP Address	Serv Port	Slave Node	Function Code	Device Modbus Address
fc1	Code	Register (bit)	1/10th Seconds	Bit Count	0	IP Address	Port #	Address	Read Coil (0x)	Register
fc2	Code	Register (bit)	1/10th Seconds	Bit Count	0	IP Address	Port #	Address	Read Input (1x)	Register
fc3	Code	Register	1/10th Seconds	Word Count	Code	IP Address	Port #	Address	Read Holding Registers (4x)	Register
fc4	Code	Register	1/10th Seconds	Word Count	0	IP Address	Port #	Address	Read Input Registers (3x)	Register
fc5	Code	1 bit	1/10th Seconds	Bit Count	0	IP Address	Port #	Address	Force (Write) Single Coil (0x)	Register
fc6	Code	1 bit	1/10th Seconds	Word Count	0	IP Address	Port #	Address	Preset (Write) Single Register (4x)	Register
fc15	Code	Register (bit)	1/10th Seconds	Bit Count	0	IP Address	Port #	Address	Force (Write) Multiple Coil (0x)	Register
fc16	Code	Register	1/10th Seconds	Word Count	0	IP Address	Port #	Address	Preset (Write) Multiple Register (4x)	Register

The first part of the record is the module Information, which relates to the MVI46 module and the second part contains information required to interface to the Server device.

Command list example:



Enable

Yes, No

This field defines whether the command is to be executed and under what conditions.

Value	Description
No	The command is disabled and will not be executed in the normal polling sequence.
Yes	The command is executed each scan of the command list if the Poll Interval Time is set to zero. If the Poll Interval time is set, the command will be executed, when the interval timer expires.

Internal Address

0 to 4999 (for word-level addressing)

or

0 to 65535 (for bit-level addressing)

This field specifies the database address in the module's internal database to use as the destination for data brought in by a read command or as the source for data to be sent out by a write command. The database address is interpreted as a bit address or a 16-bit word (register) address, depending on the Modbus Function Code used in the command.

- For Modbus functions 1, 2, 5, and 15, this parameter is interpreted as a bit-level address.
- For Modbus functions 3, 4, 6, and 16, this parameter is interpreted as a word- or register-level address.

Poll Interval

0 to 65535

This parameter specifies the minimum interval to execute continuous commands (Enable code of 1). The parameter is entered in tenths of a second. Therefore, if a value of 100 is entered for a command, the command executes no more frequently than every 10 seconds.

Reg Count

Regs: 1 to 125

Coils: 1 to 800

This parameter specifies the number of 16-bit registers or binary bits to be transferred by the command.

- Functions 5 and 6 ignore this field as they apply only to a single data point.
- For functions 1, 2, and 15, this parameter sets the number of bits (inputs or coils) to be transferred by the command.
- For functions 3, 4, and 16, this parameter sets the number of registers to be transferred by the command.

Swap Code

No Change

Word Swap

Word and Byte Swap

Byte Swap

This parameter defines if and how the order of bytes in data received or sent is to be rearranged. This option exists to allow for the fact that different manufacturers store and transmit multi-byte data in different combinations that do other manufacturers. This parameter is helpful when dealing with floating-point or other multi-byte values, as there is no one standard method of storing these data types. This parameter can be set to rearrange the byte order of data received or sent into order more useful or convenient for other applications. The following table defines the valid *Swap Code* values and the effect they have on the byte-order of the data.

Swap Code	Description
None	No Change is made in the byte ordering (1234 = 1234)
Swap Words	The words are swapped (1234=3412)
Swap Words & Bytes	The words are swapped then the bytes in each word are swapped (1234=4321)
Swap Bytes	The bytes in each word are swapped (1234=2143)

These swap operations affect 4-byte (or 2-word) groups of data. Therefore, data swapping using these Swap Codes should be done only when using an even number of words, such as when 32-bit integer or floating-point data is involved.

Node IP Address

xxx.xxx.xxx.xxx

The IP address of the device being addressed by the command.

Service Port

502 or other supported ports on server

Use a value of 502 when addressing Modbus TCP/IP servers that are compatible with the Schneider Electric MBAP specifications (this will be most devices). If a server implementation supports another service port, enter the value here.

Slave Address

0 - Broadcast to all nodes

1 to 255

Use this parameter to specify the slave address of a remote Modbus Serial device through a Modbus Ethernet to Serial converter.

Note: Use the Node IP Address parameter (page 31) to address commands to a remote Modbus TCP/IP device.
Note: Most Modbus devices accept an address in the range of only 1 to 247, so check with slave device manufacturer to see if a particular slave can use addresses 248 to 255.
If the value is set to zero, the command will be a broadcast message on the network. The Modbus protocol permits broadcast commands for **write** operations. **Do not** use node address 0 for **read** operations.

Modbus Function

1, 2, 3, 4, 5, 6, 15, or 16

This parameter specifies the Modbus Function Code to be executed by the command. These function codes are defined in the Modbus protocol. The following table lists the purpose of each function supported by the module. More information on the protocol is available from www.modbus.org.

Modbus Function Code	Description
1	Read Coil Status
2	Read Input Status
3	Read Holding Registers
4	Read Input Registers
5	Force (Write) Single Coil
6	Preset (Write) Single Register
15	Force Multiple Coils
16	Preset Multiple Registers

MB Address in Device

This parameter specifies the starting Modbus register or bit address in the slave to be used by the command. Refer to the documentation of each Modbus slave device for the register and bit address assignments valid for that device.

The Modbus Function Code determines whether the address will be a register- or bit- level OFFSET address into a given data type range. The offset will be the target data address in the slave minus the base address for that data type. Base addresses for the different data types are:

- 00001 or 000001 (0x0001) for bit-level Coil data (Function Codes 1, 5, and 15).
- 10001 or 100001 (1x0001) for bit-level Input Status data (Function Code 2)
- 30001 or 300001 (3x0001) for Input Register data (Function Code 4)
- 40001 or 400001 (4x0001) for Holding Register data (Function Codes 3, 6, and 16).

Address calculation examples:

- For bit-level Coil commands (FC 1, 5, or 15) to read or write a Coil 0X address 00001, specify a value of 0 (00001 - 00001 = 0).
- For Coil address 00115, specify 114

  (00115 - 00001 = 114)
- For register read or write commands (FC 3, 6, or 16) 4X range, for 40001, specify a value of 0

  (40001 - 40001 = 0).
- For 01101, 11101, 31101 or 41101, specify a value of 1100.

  (01101 - 00001 = 1100)
  (11101 -10001 = 1100)
  (31101 - 30001 = 1100)
  (41101 - 40001 = 1100)

Note: If the documentation for a particular Modbus slave device lists data addresses in hexadecimal (base16) notation, you will need to convert the hexadecimal value to a decimal value to enter in this parameter. In such cases, it is not usually necessary to subtract 1 from the converted decimal number, as this addressing scheme typically uses the exact offset address expressed as a hexadecimal number.

Comment

0 to 35 alphanumeric characters

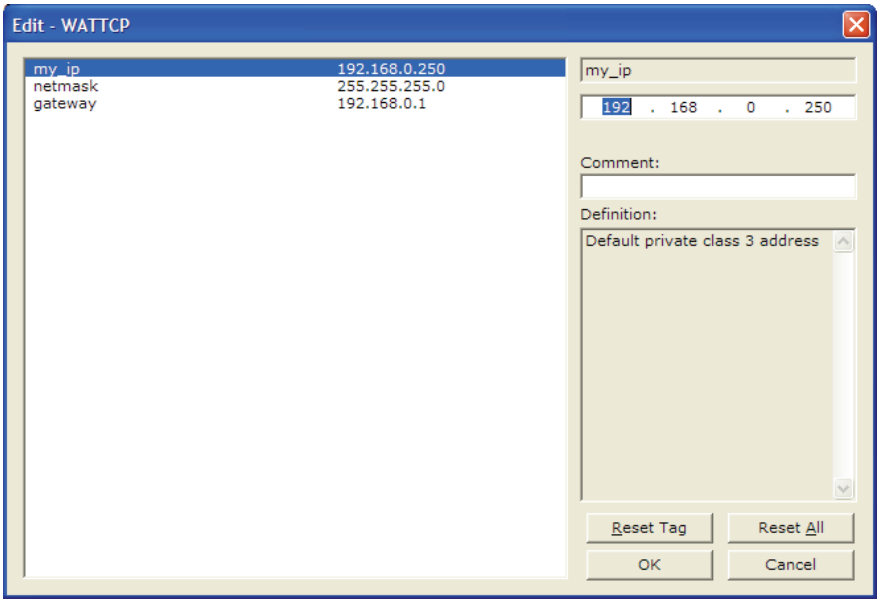
2.1.7 Ethernet Configuration

Use this procedure to configure the Ethernet settings for your module. You must assign an IP address, subnet mask and gateway address. After you complete this step, you can connect to the module with an Ethernet cable.

1 Determine the network settings for your module, with the help of your network administrator if necessary. You will need the following information:
   - IP address (fixed IP required) _____ . _____ . _____ . _____
   - Subnet mask _____ . _____ . _____ . _____
   - Gateway address _____ . _____ . _____ . _____

Note: The Gateway Address is optional, and is not required for networks that do not use a default gateway.

2 Double-click the **ETHERNET CONFIGURATION** icon. This action opens the **EDIT** dialog box.



3 Edit the values for my_ip, netmask (subnet mask) and gateway (default gateway).
4 When you are finished editing, click **OK** to save your changes and return to the ProSoft Configuration Builder window.

2.2 Setting Command Control Bits

Toggle the CMD_BITS_CNTRL bit in the ladder logic (Rung 6 of LAD4 - CNTRL in the sample program) to execute the Modbus Commands as structured in the MNET.CFG file. This will cause continuous traffic on the Ethernet network with data flow as configured in MNET.CFG file.

This rung also selects the Commands to enable as set by the Enabled bits in each word N50:21 through N50:50 representing each client.

- Client-0 enable Command bits are in N50:21, MSB xxxx xxxx xxxx xxxx LSB.
- Client-1 enable Command bits are in N50:22
- Client-29 enable Command bits are in N50:50

To change the set of Clients and their Commands to execute, change the bit setting from 0 to 1 or from 1 to 0 then toggle the CMD_BITS_CNTRL bit.

Example: Set Client-1 in N50:22 as follows: MSB 0001 0000 0100 0100 LSB. Bits 2 and 6 and 12 have been enabled. This means that Client-1 commands indexed as 2, 6 and 12 will be executed.

N50:20 contains Value 6000 to trigger the Block 6000, the Command Control Bits function.

2.3 Configuring the Floating Point Data Transfer

A common question when using the module as a Modbus TCP/IP client is how floating point data is handled. This really depends on the server device and how it addresses this application.

Just because your application is reading/writing floating point data, does not mean that you must configure the Float Flag, Float Start parameters within the module.

These parameters are only used to support what is typically referred to as Enron or Daniel Modbus, where one register address must have 32 bits, or one floating point value. Below is an example:

Example #1

Modbus Address	Data Type	Parameter
47101	32 bit REAL	TEMP Pump #1
47102	32 bit REAL	Pressure Pump #1
47103	32 bit REAL	TEMP Pump #2
47104	32 bit REAL	Pressure Pump #2

With the module configured as a master, you only need to enable these parameters to support a write to this type of addressing (Modbus FC 6 or 16).

If the slave device shows addressing as shown in Example #2, then you need not do anything with the Float Flag, Float Start parameters, as they use two Modbus addresses to represent one floating point value:

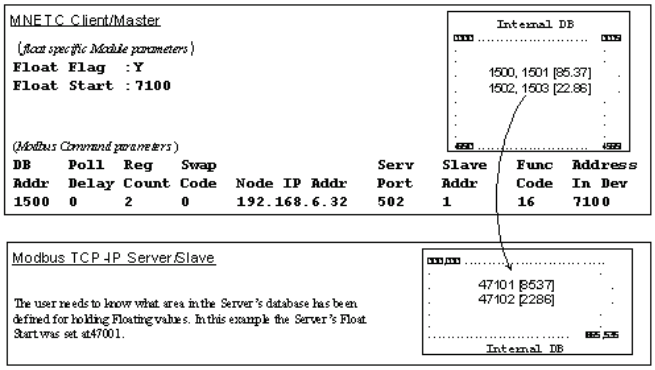
Example #2

Modbus Address	Data Type	Parameter
47101	32 bit REAL	TEMP Pump #1
47103	32 bit REAL	Pressure Pump #1
47105	32 bit REAL	TEMP Pump #2
47107	32 bit REAL	Pressure Pump #2

Because each 32 bit REAL value is represented by two Modbus Addresses (example 47101 and 47102 represent TEMP Pump #1), then you need not set the Float Flag, or Float Start for the module for Modbus FC 6 or 16 commands being written to the slave.

Below are specific examples:

Master is issuing Modbus command with FC 16 (with Float Flag: Yes) to transfer Float data to Server.



(Float specific module parameters)

Float Flag: "Y" tells the Client to consider the data values that need to be sent to the Server as floating point data where each data value is composed of 2 words (4 bytes or 32 bits).

Float Start: Tells the Client that if this address number is <= the address number in "Addr in Dev" parameter to double the byte count quantity to be included in the Command FC6 or FC16 to be issued to the Server. Otherwise the Client will ignore the "Float Flag: Y" and treat data as composed of 1 word, 2 bytes.

(Modbus Command Parameters)

DB Addr - Tells the Client where in its data memory is the beginning of data to obtain and write out to the Server (slave) device.

Reg Count - Tells the Client how many data points to send to the Server. Two counts will mean two floating points with Float Flag: Y and the "Addr in Dev" => the "Float Start" Parameter.

Swap Code - Tells the Client how to orient the Byte and Word structure of the data value. This is device dependent. Check Command Entry formats Section.

Func Code - Tells the Client to write the float values to the Server. FC16.

Addr in Dev - Tells the Client where in the Server's database to locate the data.

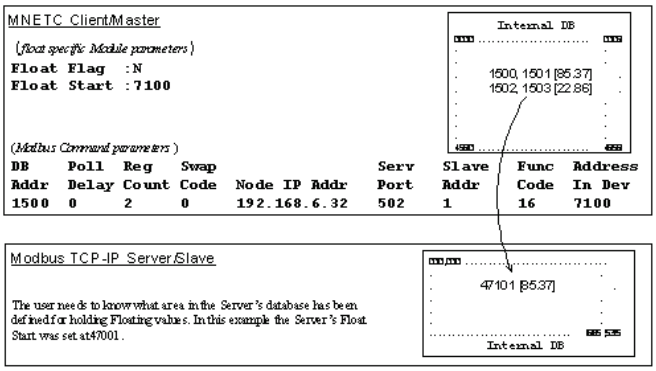
In the above example, the Client's Modbus command to transmit inside the Modbus/TCP packet will be as follows.

	Slave address	Function Code	Address in Device	Reg count	Byte Count	Data	
DEC	01	16	7100	2	8	85.37	22.86
HEX	01	10	1B BC	00 02	08	BD 71 42 AA	E1 48 41 B6

In conclusion

The Client's Modbus TCP/IP packet contains the data byte and data word counts that have been doubled from the amount specified by Reg Count due to the Float flag set to Y. Some Servers look for the byte count in the data packet to know the length of the data to read from the wire. Other servers know at which byte the data begins and read from the wire the remaining bytes in the packet as the data the Client is sending.

Client is issuing Modbus command with FC 16 (with Float Flag: No) to transfer Float data.



Float Flag: "N" tells the Client to ignore the floating values and treat each register data as a data point composed of 1 word, 2 bytes or 16 bits.

Float Start: Ignored.

DB Addr - same as when Float Flag: Y.

Reg Count - Tells the Client how many data points to send to the Server.

Swap Code - same as when Float Flag: Y.

Func Code - same as when Float Flag: Y.

Addr in Dev - same as when Float Flag: Y as long as the Server's Float Flag = Y.

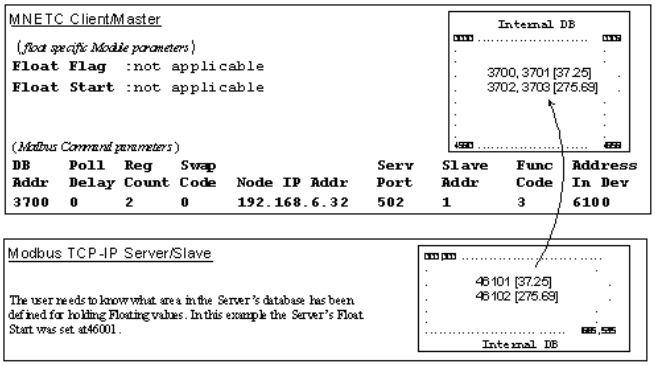
In the above example, the Client's Modbus command to transmit inside the Modbus/TCP pocket will be as follows.

	Slave address	Function Code	Address in Device	Reg count	Byte Count	Data
DEC	01	16	7100	2	4	85.37
HEX	01	10	1B BC	00 02	04	BD 71 42 AA

In conclusion

The Client's Modbus TCP/IP packet contains the data byte and data word counts that have NOT been doubled from the amount specified by Reg Count due to the Float Flag set to N. The Server looks for the byte count in the data packet to know the length of the data to read from the wire. Because of insufficient byte count, some servers will read only half the data from the Client's transmission. Other servers will read all 8 bytes in this example because they will know where in the packet the data starts and ignore the byte count parameter inside the Modbus TCP/IP packet.

Client is issuing Modbus command with FC 3 to transfer Float data from Server.


MNETC Client/Master
(float specific Module parameters)
Float Flag :not applicable
Float Start :not applicable
(Modbus Command parameters)
DB Addr | Poll Delay | Reg Count | Swap Code | Node IP Addr | Serv Port | Slave Addr | Func Code | Address In Dev
3700 | 0 | 2 | 0 | 192.168.6.32 | 502 | 1 | 3 | 6100
Internal DB
3700, 3701 [37.25]
3702, 3703 [275.69]
Modbus TCP-IP Server/Slave
The user needs to know what area in the Server's database has been defined for holding Floating values. In this example the Server's Float Start was set at 46001.
46101 [37.25]
46102 [275.69]
Internal DB

Float Flag: Not applicable with Modbus Function Code 3.

Float Start: Not applicable with Modbus Function Code 3.

DB Addr - Tells the Client where in its data memory to store the data obtained from the Server.

Reg Count - Tells the Client how many registers to request from the Server.

Swap Code - same as above.

Func Code - Tells the Client to read the register values from the Server. FC3.

Addr in Dev - Tells the Client where in the Server's database to obtain the data.

In the above example, the Client's Modbus command to transmit inside the Modbus/TCP pocket will be as follows.

	Slave address	Function Code	Address in Device	Reg count
DEC	01	3	6100	2
HEX	01	03	17 D4	00 02

In the above example the (Enron/Daniel supporting) Slave's Modbus command to transmit inside the Modbus/TCP packet will be as follows.

	Slave address	Function Code	Byte Count	Data	
DEC	01	3	8	32.75	275.69
HEX	01	03	08	00 00 42 03	D8 52 43 89

In the above example the (a NON-Enron/Daniel supporting) Slave's Modbus command that will be transmitted inside the Modbus/TCP packet will be as follows.

	Slave address	Function Code	Byte Count	Data
DEC	01	3	4	32.75
HEX	01	03	04	00 00 42 03

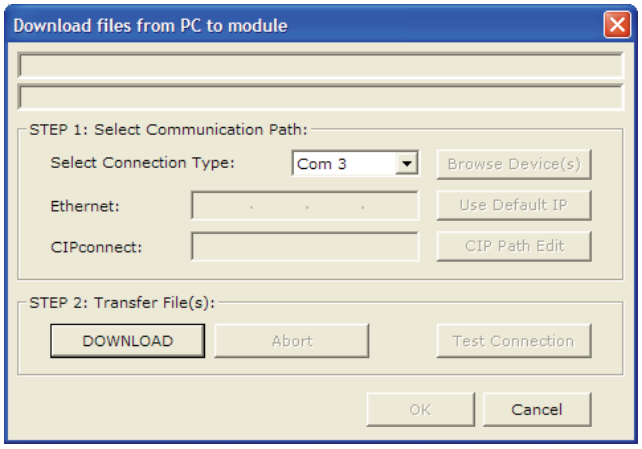
Note: You can use the Debug window to view the floating points in the Module's database. First, you must copy the registers in the F8-FLOAT file in the SLC to the N7-INTEGER file registers, and then swap the two words that make up a floating point. Then, copy the registers to the module's M1 file.
For example:
Copy F8:0 to N7:0 [length 2].
Next two move functions perform swapping.
Move N7:0 to N7:11
Move N7:1 to N7:10
The Copy function transfers the two words to the Module.
Copy N7:10 to M1:1.0 [length 2]

2.4 Download the Project to the Module

In order for the module to use the settings you configured, you must download (copy) the updated Project file from your PC to the module.

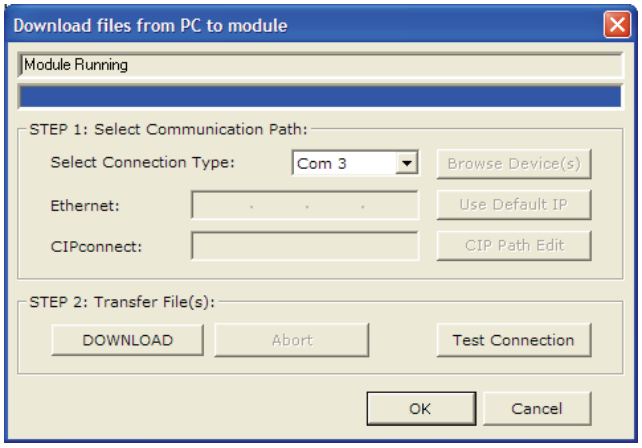
To Download the Project File

1. In the tree view in ProSoft Configuration Builder, click once to select the MVI46-MNETC module.
2. Open the **PROJECT** menu, and then choose **MODULE / DOWNLOAD.** The program will scan your PC for a valid com port (this may take a few seconds). When PCB has found a valid com port, the **DOWNLOAD** dialog box will open.



3. Choose the com port to use from the dropdown list, and then click the **DOWNLOAD** button.

   The module will perform a platform check to read and load its new settings. When the platform check is complete, the status bar in the **DOWNLOAD** dialog box with the message *"Module Running*".



3 Ladder Logic

In This Chapter

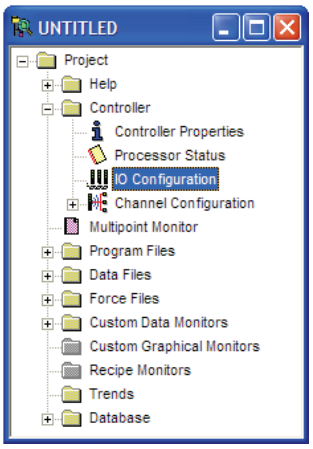
❖ Adding the Module to an Existing Project 41

Ladder logic is required for application of the MVI46-MNETC module. Tasks that must be handled by the ladder logic are module data transfer, special block handling, and status data receipt. Additionally, a power-up handler may be needed to handle the initialization of the module's data and to clear any processor fault conditions.

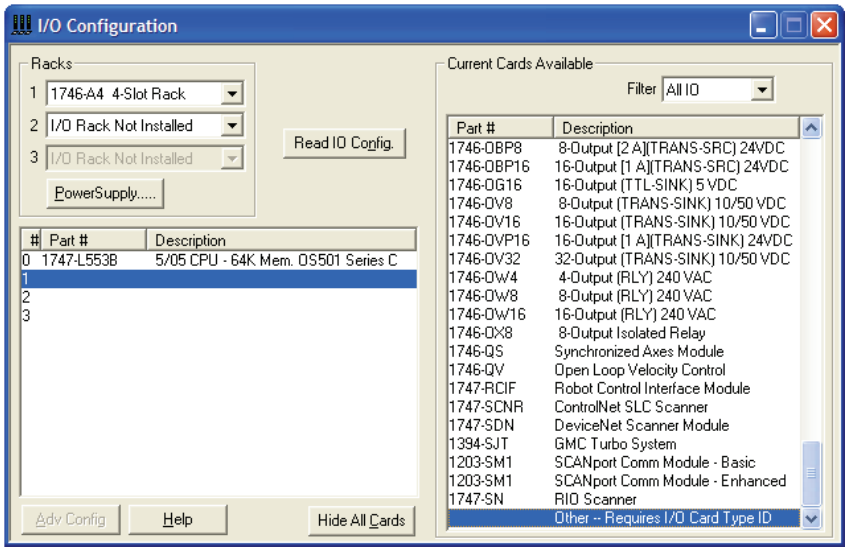
The sample ladder logic, on the ProSoft Solutions CD-ROM, is extensively commented, to provide information on the purpose and function of each rung. For most applications, the sample ladder will work without modification.

3.1 Adding the Module to an Existing Project

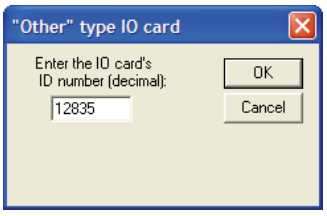
1 Add the MVI46-MNETC **module to the project.** Double-click on the I/O Configuration option in the Controller Organization window.



2 This action opens the I/O Configuration dialog box. Select an empty slot in the left pane, and then scroll to the bottom of the right pane.

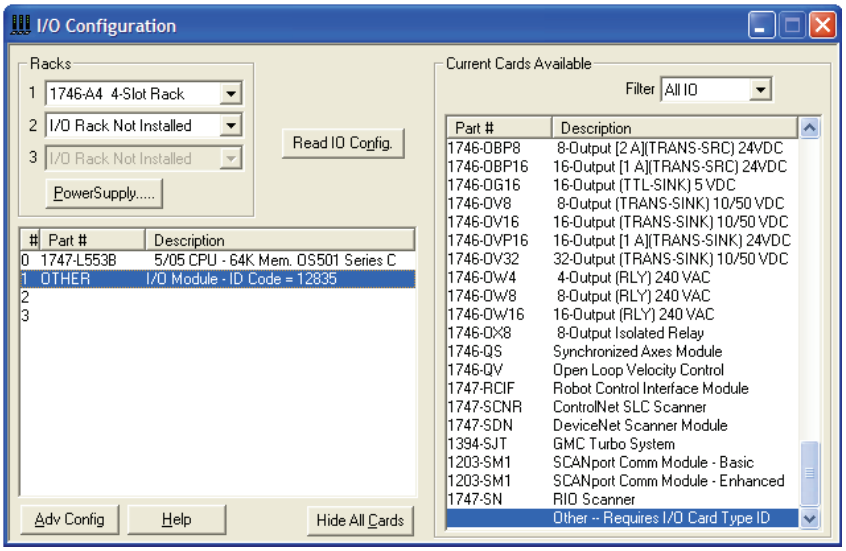

I/O Configuration
Racks
1 1746-A4 4-Slot Rack
2 I/O Rack Not Installed
3 I/O Rack Not Installed
PowerSupply.....
Read IO Config.
Part # Description
0 1747-L553B 5/05 CPU - 64K Mem. OS501 Series C
1
2
3
Adv Config
Help
Hide All Cards
Current Cards Available
Filter All IO
Part # Description
1746-OBP8 8-Output [2 A](TRANS-SRC) 24VDC
1746-OBP16 16-Output [1 A](TRANS-SRC) 24VDC
1746-OG16 16-Output (TTL-SINK) 5 VDC
1746-OV8 8-Output (TRANS-SINK) 10/50 VDC
1746-OV16 16-Output (TRANS-SINK) 10/50 VDC
1746-OVP16 16-Output [1 A](TRANS-SINK) 24VDC
1746-OV32 32-Output (TRANS-SINK) 10/50 VDC
1746-OW4 4-Output (RLY) 240 VAC
1746-OW8 8-Output (RLY) 240 VAC
1746-OW16 16-Output (RLY) 240 VAC
1746-OX8 8-Output Isolated Relay
1746-QS Synchronized Axes Module
1746-QV Open Loop Velocity Control
1747-RCIF Robot Control Interface Module
1747-SCNR ControlNet SLC Scanner
1747-SDN DeviceNet Scanner Module
1394-SJT GMC Turbo System
1203-SM1 SCANport Comm Module - Basic
1203-SM1 SCANport Comm Module - Enhanced
1747-SN RIO Scanner
Other -- Requires I/O Card Type ID

3 In the right pane, double-click **Other -- Requires I/O Card Type ID.** This action opens the **"Other" type IO card** dialog box.

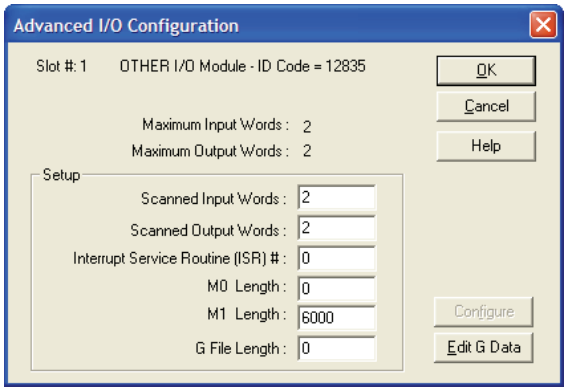


4 The module's I/O card ID number is 12835. Enter that value in the ID number field, and then click OK to dismiss the dialog box.

5 Observe that the module you selected is now in the list in the left pane of the I/O Configuration dialog box.


I/O Configuration
Racks
1 1746-A4 4-Slot Rack
2 I/O Rack Not Installed
3 I/O Rack Not Installed
PowerSupply.....
Read IO Config.
Part # Description
0 1747-L553B 5/05 CPU - 64K Mem. OS501 Series C
1 OTHER I/O Module - ID Code = 12835
2
3
Adv Config
Help
Hide All Cards
Current Cards Available
Filter All IO
Part # Description
1746-OBP8 8-Output (2 A)(TRANS-SRC) 24VDC
1746-OBP16 16-Output (1 A)(TRANS-SRC) 24VDC
1746-OG16 16-Output (TTL-SINK) 5 VDC
1746-OV8 8-Output (TRANS-SINK) 10/50 VDC
1746-OV16 16-Output (TRANS-SINK) 10/50 VDC
1746-OVP16 16-Output (1 A)(TRANS-SINK) 24VDC
1746-OV32 32-Output (TRANS-SINK) 10/50 VDC
1746-OW4 4-Output (RLY) 240 VAC
1746-OW8 8-Output (RLY) 240 VAC
1746-OW16 16-Output (RLY) 240 VAC
1746-OX8 8-Output Isolated Relay
1746-QS Synchronized Axes Module
1746-QV Open Loop Velocity Control
1747-RCIF Robot Control Interface Module
1747-SCNR ControlNet SLC Scanner
1747-SDN DeviceNet Scanner Module
1394-SJT GMC Turbo System
1203-SM1 SCANport Comm Module - Basic
1203-SM1 SCANport Comm Module - Enhanced
1747-SN RIO Scanner
Other -- Requires I/O Card Type ID

6 Select and double-click the new module in the left pane. This action opens the Advanced I/O Configuration dialog box. Fill in the dialog box with the values shown in the following illustration.



Field	Value
Scanned Input Words	2
Scanned Output Words	2
Interrupt Service Routine (ISR)#	0
M0 Length	0
M1 Length	6000
G File Length	0

7 Click OK to save your configuration.

8 Copy the Ladder Logic and data files from the sample program and paste them into your existing program.

Important: Take care not to overwrite existing data files in your application with data files in the sample application. Rename either the source or the destination data files, and then search and replace references in the ladder for instances of any renamed files.

9 Save and Download the new application to the controller and place the processor in run mode.

4 Diagnostics and Troubleshooting

In This Chapter

❖ Reading Status Data from the Module 45
❖ LED Status Indicators 56

The module provides information on diagnostics and troubleshooting in the following forms:

- Status data values are transferred from the module to the processor
- Data contained in the module can be viewed through the Configuration/Debug port attached to a terminal emulator
- LED status indicators on the front of the module provide information on the module's status

4.1 Reading Status Data from the Module

The MVI46-MNETC module returns status data words that can be used to determine the module's operating status. This data is located in the module's database at a user set location. This data can be transferred to the SLC processor continuously or at interval set by the user in ladder logic.

The Configuration/Debug port provides the following functionality:

- Full view of the module's configuration data
- View of the module's status data
- Complete display of the module's internal database (registers 0 to 3999)
- Version Information
- Control over the module (warm boot, cold boot, transfer configuration)
- Facility to upload and download the module's configuration file

4.1.1 Required Hardware

You can connect directly from your computer's serial port to the serial port on the module to view configuration information, perform maintenance, and send (upload) or receive (download) configuration files.

ProSoft Technology recommends the following minimum hardware to connect your computer to the module:

- 80486 based processor (Pentium preferred)
- 1 megabyte of memory
- At least one UART hardware-based serial communications port available. USB-based virtual UART systems (USB to serial port adapters) often do not function reliably, especially during binary file transfers, such as when uploading/downloading configuration files or module firmware upgrades.

4.1.2 The Configuration/Debug Menu

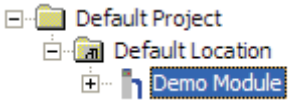
The Configuration and Debug menu for this module is arranged as a tree structure, with the Main Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the Main menu.

Because this is a text-based menu system, you enter commands by typing the command letter from your computer keyboard in the diagnostic window in ProSoft Configuration Builder (PCB). The module does not respond to mouse movements or clicks. The command executes as soon as you press the command letter — you do not need to press **[ENTER].** When you type a command letter, a new screen will be displayed in your terminal application.

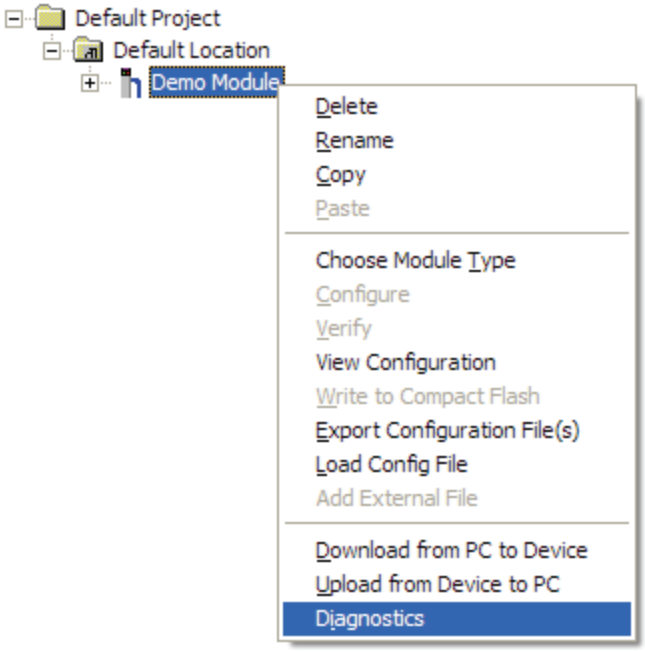
Using the Diagnostic Window in ProSoft Configuration Builder

To connect to the module's Configuration/Debug serial port,

1 Start PCB, and then select the module to test. Click the right mouse button to open a shortcut menu.

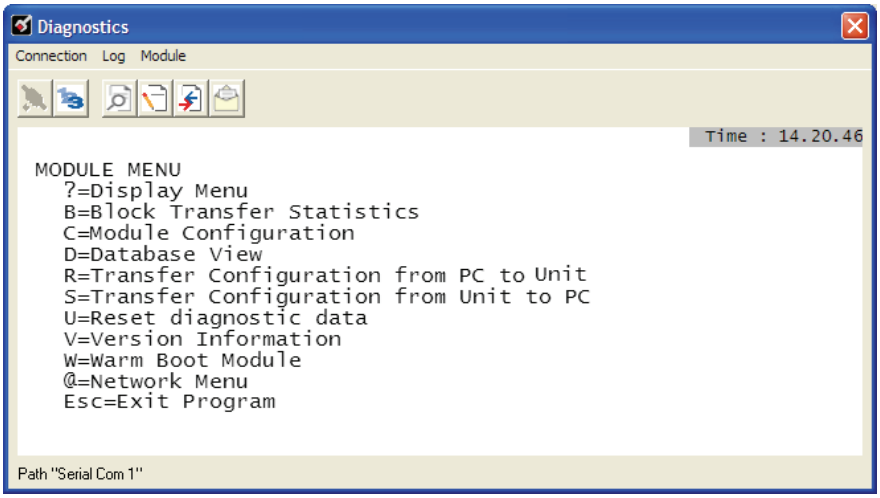


2 On the shortcut menu, choose **DIAGNOSTICS.**



This action opens the **DIAGNOSTICS** dialog box.

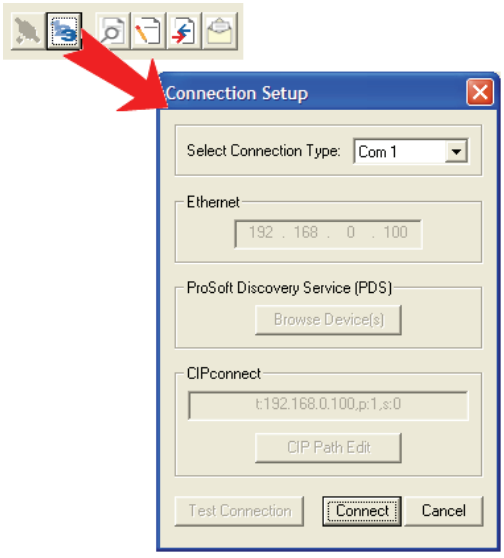
3 Press **[?]** to open the Main Menu.



Important: The illustrations of configuration/debug menus in this section are intended as a general guide, and may not exactly match the configuration/debug menus in your own module.

If there is no response from the module, follow these steps:

1 Click to configure the connection. On the Connection Setup dialog box, select a valid com port or other connection type supported by the module.



2 Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.

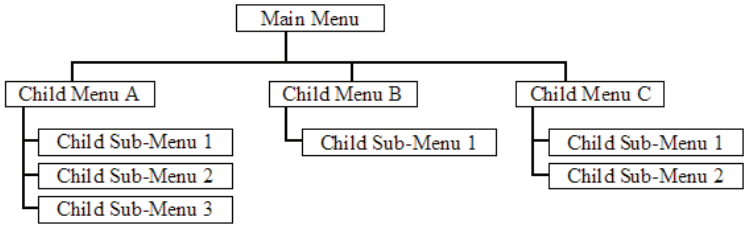
3 On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

If you are still not able to establish a connection, contact ProSoft Technology for assistance.

Navigation

All of the sub-menus for this module contain commands to redisplay the menu or return to the previous menu. You can always return from a sub-menu to the next higher menu by pressing **[M]** on your keyboard.

The organization of the menu structure is represented in simplified form in the following illustration:



The remainder of this section shows you the menus available for this module, and briefly discusses the commands available to you.

Keystrokes

The keyboard commands on these menus are almost always non-case sensitive. You can enter most commands in lower case or capital letters.

The menus use a few special characters (**[?], [-], [+], [@]**) that must be entered exactly as shown. Some of these characters will require you to use the **[SHIFT], [CTRL]** or **[ALT]** keys to enter them correctly. For example, on US English keyboards, enter the **[?]** command as **[SHIFT][/].**

Also, take care to distinguish capital letter **[I]** from lower case letter **[L]** (L) and number **[1];** likewise for capital letter **[O]** and number **[0].** Although these characters look nearly the same on the screen, they perform different actions on the module.

4.1.3 Main Menu

When you first connect to the module from your computer, your terminal screen will be blank. To activate the main menu, press the **[?]** key on your computer's keyboard. If the module is connected properly, the following menu will appear.

```
MVI46-MNETC COMMUNICATION MODULE MENU
  ?=Display Menu
  B=Block Transfer Statistics
  C=Module Configuration
  D=Modbus Database View
  E=Client Command List Errors
  I=Client Command List
  R=Transfer Configuration from PC to MVI Unit
  S=Transfer Configuration from MVI Unit to PC
  U=Reset diagnostic data
  V=Version Information
  W=Warm Boot Module
  0=Client Communication Status
  5=Client Configuration
  @=Network Menu          Esc=Exit Program
```

Caution: Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other failures. Only use these commands if you are specifically directed to do so by ProSoft Technology Technical Support staff. Some of these command keys are not listed on the menu, but are active nevertheless. Please be careful when pressing keys so that you do not accidentally execute an unwanted command.

Viewing Block Transfer Statistics

Press **[B]** from the Main Menu to view the Block Transfer Statistics screen.

Use this command to display the configuration and statistics of the backplane data transfer operations between the module and the processor. The information on this screen can help determine if there are communication problems between the processor and the module.

Tip: To determine the number of blocks transferred each second, mark the numbers displayed at a specific time. Then some seconds later activate the command again. Subtract the previous numbers from the current numbers and divide by the quantity of seconds passed between the two readings.

Viewing Module Configuration

Press **[C]** to view the Module Configuration screen.

Use this command to display the current configuration and statistics for the module.

Opening the Database Menu

Press **[D]** to open the Database View menu. Use this menu command to view the current contents of the module's database.

Opening the Command List Menu

Press **[L]** to open the Command List menu. Use this command to view the configured command list for the module.

Opening the Command Error List Menu

Press **[I]** to open the Command Error List. This list consists of multiple pages of command list error/status data. Press **[?]** to view a list of commands available on this menu.

Receiving the Configuration File

Press **[R]** to download (receive) the current configuration file from the module. For more information on receiving and sending configuration files, please see Uploading and Downloading the Configuration File.

Sending the Configuration File

Press **[S]** to upload (send) an updated configuration file to the module. For more information on receiving and sending configuration files, please see Uploading and Downloading the Configuration File.

Resetting diagnostic data

Press **[U]** to reset the status counters for the client and/or servers in the module.

Viewing Version Information

Press **[V]** to view Version information for the module.

Use this command to view the current version of the software for the module, as well as other important values. You may be asked to provide this information when calling for technical support on the product.

Values at the bottom of the display are important in determining module operation. The Program Scan Counter value is incremented each time a module's program cycle is complete.

Tip: Repeat this command at one-second intervals to determine the frequency of program execution.

Warm Booting the Module

Caution: Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other failures. Only use these commands if you are specifically directed to do so by ProSoft Technology Technical Support staff. Some of these command keys are not listed on the menu, but are active nevertheless. Please be careful when pressing keys so that you do not accidentally execute an unwanted command.

Press **[W]** from the Main Menu to warm boot (restart) the module. This command will cause the program to exit and reload, refreshing configuration parameters that must be set on program initialization. Only use this command if you must force the module to re-boot.

Viewing Client Status

Press **[0]** (zero) to display the statistics of the client.

Viewing Client Configuration

Press **[5]** to display the configuration information for the client.

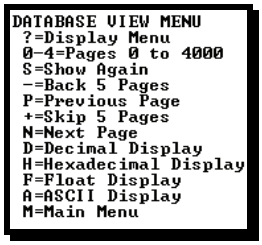
Exiting the Program

Caution: Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other failures. Only use these commands if you are specifically directed to do so by ProSoft Technology Technical Support staff. Some of these command keys are not listed on the menu, but are active nevertheless. Please be careful when pressing keys so that you do not accidentally execute an unwanted command.

Press **[Esc]** to restart the module and force all drivers to be loaded. The module will use the configuration stored in the module's Flash memory to configure the module.

4.1.4 Modbus Database View

Press **[D]** to open the Modbus Database View menu. Use this command to view the module's internal database values. Press **[?]** to view a list of commands on this menu.



All data contained in the module's database is available for viewing using the commands. Refer to Modbus Protocol Specification for information on the structure of Modbus messages. Each option available on the menu is discussed in the following topics.

Viewing Register Pages

To view sets of register pages, use the keys described below:

Command	Description
[0]	Display registers 0 to 99
[1]	Display registers 1000 to 1099
[2]	Display registers 2000 to 2099

And so on. The total number of register pages available to view depends on your module's configuration.

Redisplaying the Current Page

Press **[S]** to display the current page of data.

Moving Back Through 5 Pages of Registers

Press **[-]** from the Database View menu to skip five pages back in the database to see the previous 100 registers of data.

Viewing the Previous 100 Registers of Data

Press **[P]** from the Database View menu to display the previous 100 registers of data.

Moving Forward Through 5 Pages of Registers

Press **[+]** from the Database View menu to skip five pages ahead in the database to see the next 100 registers of data.

Viewing the Next 100 Registers of Data

Press **[N]** from the Database View menu to select and display the next 100 registers of data.

Viewing Data in Decimal Format

Press **[D]** to display the data on the current page in decimal format.

Viewing Data in Hexadecimal Format

Press **[H]** to display the data on the current page in hexadecimal format.

Viewing Data in Floating Point Format

Press **[F]** from the Database View menu. Use this command to display the data on the current page in floating point format. The program assumes that the values are aligned on even register boundaries. If floating-point values are not aligned as such, they are not displayed properly.

Viewing Data in ASCII (Text) Format

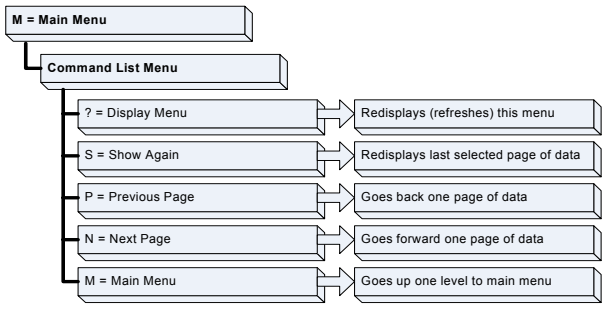
Press **[A]** to display the data on the current page in ASCII format. This is useful for regions of the database that contain ASCII data.

Returning to the Main Menu

Press **[M]** to return to the Main Menu.

4.1.5 Command List Menu

Use this menu to view the configured command list for the module. Press **[?]** to view a list of commands available on this menu.


M = Main Menu
Command List Menu
? = Display Menu
Redisplays (refreshes) this menu
S = Show Again
Redisplays last selected page of data
P = Previous Page
Goes back one page of data
N = Next Page
Goes forward one page of data
M = Main Menu
Goes up one level to main menu

Redisplaying the Menu

Press **[?]** to display the current menu. Use this command when you are looking at a screen of data, and want to view the menu choices available to you.

Viewing the Previous Page of Commands

Press **[P]** to display the previous page of commands.

Viewing the Next Page of Commands

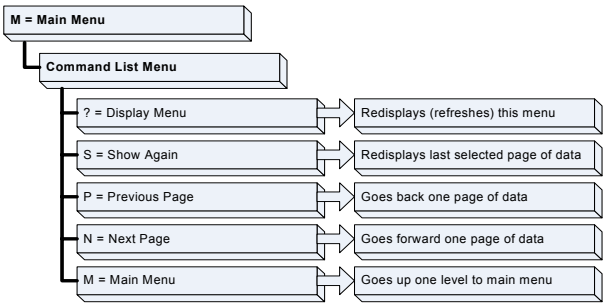
Press **[N]** to display the next page of commands.

Returning to the Main Menu

Press **[M]** to return to the Main Menu.

4.1.6 Master Command Error List Menu

Use this menu to view the command error list for the module. Press **[?]** to view a list of commands available on this menu.


M = Main Menu
Command List Menu
? = Display Menu
Redisplays (refreshes) this menu
S = Show Again
Redisplays last selected page of data
P = Previous Page
Goes back one page of data
N = Next Page
Goes forward one page of data
M = Main Menu
Goes up one level to main menu

Redisplaying the Menu

Press **[?]** to display the current menu. Use this command when you are looking at a screen of data, and want to view the menu choices available to you.

Viewing the Previous Page of Commands

Press **[P]** to display the previous page of commands.

Viewing the Next Page of Commands

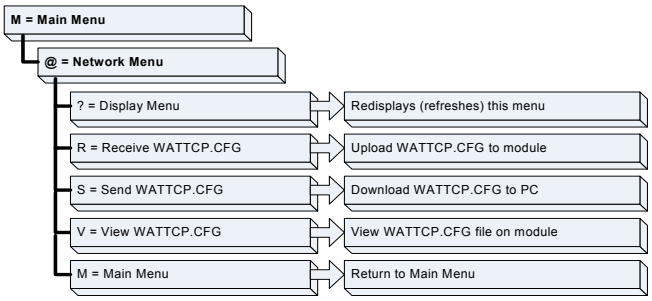
Press **[N]** to display the next page of commands.

Returning to the Main Menu

Press **[M]** to return to the Main Menu.

4.1.7 Network Menu

The network menu allows you to send, receive, and view the WATTCP.CFG file that contains the IP and gateway addresses, and other network information.


M = Main Menu
@ = Network Menu
? = Display Menu
Redisplays (refreshes) this menu
R = Receive WATTCP.CFG
Upload WATTCP.CFG to module
S = Send WATTCP.CFG
Download WATTCP.CFG to PC
V = View WATTCP.CFG
View WATTCP.CFG file on module
M = Main Menu
Return to Main Menu

Transferring WATTCP.CFG to the module

Press **[R]** to transfer a new WATTCP.CFG file from the PC to the module. Use this command to change the network configuration for the module (for example, the module's IP address).

Press **[Y]** to confirm the file transfer, and then follow the instructions on the terminal screen to complete the file transfer process.

Transferring WATTCP.CFG to the PC

Press **[S]** to transfer the WATTCP.CFG file from the module to your PC.

Press **[Y]** to confirm the file transfer, and then follow the instructions on the terminal screen to complete the file transfer process.

After the file has been successfully transferred, you can open and edit the file to change the module's network configuration.

Viewing the WATTCP.CFG file on the module

Press **[V]** to view the module's WATTCP.CFG file. Use this command to confirm the module's current network settings.

```
WATTCP.CFG FILE:
# ProLinx Communication Gateways, Inc.
# Default private class 3 address
my_ip=192.168.0.75
# Default class 3 network mask
netmask=255.255.255.0
# name server 1 up to 9 may be included
# nameserver=xxx.xxx.xxx.xxx
# name server 2
# nameserver=xxx.xxx.xxx.xxx
# The gateway I wish to use
gateway=192.168.0.1
# some networks (class 2) require all three parameters
# gateway,network,subnetmask
# gateway 192.168.0.1,192.168.0.0,255.255.255.0
# The name of my network
# domainslist="mynetwork.name"
```

Returning to the Main Menu

Press **[M]** to return to the Main Menu.

4.2 LED Status Indicators

The LEDs indicate the module's operating status as follows:

Module	Color	Status	Indication
CFG	Green	On	Data is being transferred between the module and a remote terminal using the Configuration/Debug port.
		Off	No data is being transferred on the Configuration/Debug port.
P1	Green	On	Port not used
		Off	Port not used
P2	Green	On	Port not used
		Off	Port not used
APP	Amber	Off	The MVI46-MNETC is working normally.
		On	The MVI46-MNETC module program has recognized a communication error.
BP ACT	Amber	On	The LED is on when the module is performing a write operation on the backplane.
		Off	The LED is off when the module is performing a read operation on the backplane. Under normal operation, the LED should blink rapidly on and off.
OK	Red/ Green	Off	The card is not receiving any power and is not securely plugged into the rack.
		Green	The module is operating normally.
		Red	The program has detected an error or is being configured. If the LED remains red for over 10 seconds, the program has probably halted. Remove the card from the rack and re-insert the card to restart the module's program.
BAT	Red	Off	The battery voltage is OK and functioning.
		On	The battery voltage is low or battery is not present. Allow battery to charge by keeping module plugged into rack for 24 hours. If BAT LED still does not go off, contact ProSoft Technology, as this is not a user serviceable item.

If a configuration error is found for the client, the client configuration error word will have a value other than zero. The configuration error word bits have the following definitions:

Bit	Description	Value
0	Not used	0x0001
1	Not used	0x0002
2	Not used	0x0004
3	Not used	0x0008
4	Invalid retry count parameter	0x0010
5	The float flag parameter is not valid.	0x0020
6	The float start parameter is not valid.	0x0040
7	The float offset parameter is not valid.	0x0080
8	APP	0x0100
9	CMD ERR	0x0200

Bit	Description	Value
10	Not used	0x0400
11	Not used	0x0800
12	Not used	0x1000
13	Not used	0x2000
14	Not used	0x4000
15	Not used	0x8000

Correct any invalid data in the configuration for proper module operation. When the configuration contains a valid parameter set, all the bits in the configuration word will be clear. This does not indicate that the configuration is valid for the user application. Make sure each parameter is set correctly for the specific application.

If the APP, BP ACT and OK LEDs blink at a rate of every one-second, this indicates a serious problem with the module. Call ProSoft Technology support to arrange for repairs.

Clearing a Fault Condition

Typically, if the OK LED on the front of the module becomes illuminated red for over ten seconds, a hardware problem has been detected in the module or the program has exited. To attempt to clear the condition:

1. Turn the power to the rack off
2. Remove the card from the rack
3. Make certain the Compact Flash is installed and all jumpers are set correctly
4. Re-insert the card in the rack and turn the power back on
5. Verify the configuration data being transferred to the module from the SLC processor

If the module's OK LED does not turn green, make sure the module is inserted completely into the rack. If this does not cure the problem, contact ProSoft Technology.

Troubleshooting

Use the following troubleshooting steps if you encounter problems when the module is powered up. If these steps do not resolve your problem, please contact ProSoft Technology Technical Support.

Problem Description	Steps to take
Processor Fault	Confirm that the module is plugged into the slot that has been configured for the MVI46-MNETC module. Confirm that the slot in the rack configuration has been set up correctly.
Processor I/O LED flashes	This indicates there is a problem with backplane communications. Verify that this and all modules in the rack are configured in the processor.
BP ACT LED remains off or blinks slowly	This indicates that backplane transfer operations are failing. Use the Configuration/Debug port facility to check this. To establish backplane communications, verify the following items: ▪ The backplane driver is loaded in the module. ▪ The module is configured for read and write block data transfer. ▪ The ladder logic handles all read and write block situations. ▪ The module is configured in the processor.
OK LED remains red	The program has halted or a critical error has occurred. Connect to the Configuration/Debug port to see if the module is running. If the program has halted, turn off power to the rack, remove the card from the rack and re-insert the card in the rack, and then restore power to the rack.

4.2.1 Ethernet LED Indicators

LED	State	Description
Data	Off	No activity on the Ethernet port.
	Green Flash	The Ethernet port is actively transmitting or receiving data.
Link	Off	No physical network connection is detected. No Ethernet communication is possible. Check wiring and cables.
	Green Solid	Physical network connection detected. This LED must be on solid for Ethernet communication to be possible.

5 Reference

In This Chapter

- Product Specifications 59
- Functional Overview 61
- Cable Connections 73

5.1 Product Specifications

The MVI46 Modbus TCP/IP Client Communication Module allows Rockwell Automation® SLC® processors to interface easily with other Modbus compatible devices.

Compatible devices include Modicon PAC's as well as a wide variety of instruments and devices. The module acts as an input/output module between the Modbus network and the Rockwell Automation backplane. The data transfer from the processor is asynchronous from the actions on the Modbus Client controlled network. A 5000-word register space in the module exchanges data between the processor and the Modbus network.

5.1.1 General Specifications

- Single Slot - 1746 backplane compatible (Local or extended I/O rack only. Remote rack not supported)
- The module is recognized as an Input/Output module and has access to processor memory for data transfer between processor and module using M0/M1 files
- Ladder Logic is used for data transfer between module and processor. Sample ladder file included
- Configuration data obtained from configuration text file downloaded to module. Sample configuration file included

5.1.2 General Specifications - Modbus TCP/IP Client

- 10/100 MB Ethernet port
- Module I/O data memory mapping supports up to 5000 registers and is user definable
- ProSoft Configuration Builder (PCB) software supported, a Windows-based graphical user interface providing simple product and network configuration
- Sample Ladder Logic and Add-On Instructions (AOI) are used for data transfer between module and processor and module configuration
- Personality Module (non-volatile CF card) used to store configuration allowing for quick in-the-field product replacement.

5.1.3 Functional Specifications - MVI-MNETC

The MVI46-MNETC will operate on a Local or Remote rack. This module was created to improve the performance when servers are not needed on a Modbus TCP/IP network. The module supports up to 30 clients with up to 16 commands for each client, making it easy to enable/disable the commands from the ladder logic. The client command control word contains one bit for each command. This module does not support the conditional write command.

- 10/100 MB Ethernet Application port
- Supports Enron version of Modbus protocol for floating point data transactions
- PCB includes a powerful Modbus network analyzer
- Special functions (command control, event commands, status, etc.) are supported by message transfer (unscheduled) using the MSG instruction
- Configurable parameters for the client including a minimum response delay of 0 to 65535 ms and floating point support
- Supports up to 30 clients with up to 16 commands for each client
- All data mapping begins at Modbus register 40001.
- Error codes, network error counters, and port status data available in user data memory

Client Specifications

A port configured as a virtual Modbus Client device on the MVI46-MNETC module actively issues Modbus commands to other nodes on the Modbus network. Additionally, the master ports have an optimized polling characteristic that polls slaves with communication problems less frequently. The SLC processor can be programmed to control the activity on the port by actively selecting commands from the command list to execute or issuing commands directly from the ladder logic.

5.1.4 Hardware Specifications

Specification	Description
Backplane Current Load	800 ma @ 5V (from backplane)
Operating Temperature	0 to 60°C (32 to 140°F)
Storage Temperature	-40 to 85°C (-40 to 185°F)
Relative Humidity	5% to 95% (non-condensing)
Shock	30g operational, 50g non-operational
Vibration	5 g from 10150 Hz
Processor	Compatible with Rockwell Automation SLC 5/02 M0/M1 capable processors or newer
LED indicators	Module status, Backplane transfer status, Application status, Serial activity (debug port), Ethernet link and activity, and error LED status

Specification	Description
Debug/Configuration port (CFG)	
CFG Port (CFG)	RJ45 (DB-9M with supplied cable) RS-232 only No hardware handshaking
Configuration Connector	RJ45 RS-232 Connector (RJ45 to DB-9 cable shipped with unit)
Application Ports	
Ethernet Port (Ethernet Modules)	RJ45 Connector Link and activity LED indicators Electrical Isolation 1500 V rms at 50 Hz to 60 Hz for 60 s, applied as specified in section 5.3.2 of IEC 60950: 1991 Ethernet Broadcast Storm Resiliency = less than or equal to 5000 [ARP] frames-per-second and less than or equal to 5 minutes duration

5.2 Functional Overview

This section describes how the MVI46-MNETC module transfers data between itself and the processor, and how it implements the MNETC protocol.

5.2.1 General Concepts

The following discussion explains several concepts that are important for understanding the operation of the MVI46-MNETC module.

About the MODBUS/TCP Protocol

MODBUS is a widely-used protocol originally developed by Modicon in 1978. Since that time, the protocol has been adopted as a standard throughout the automation industry.

The original MODBUS specification uses a serial connection to communicate commands and data between client and server devices on a network. Later enhancements to the protocol allow communication over Ethernet networks using TCP/IP as a "wrapper" for the MODBUS protocol. This protocol is known as MODBUS/TCP.

MODBUS/TCP is a client/server protocol. The client establishes a connection to the remote server. When the connection is established, the client sends the MODBUS/TCP commands to the server. The MVI46-MNETC module simulates up to 30 clients.

Aside from the benefits of Ethernet versus serial communications (including performance, distance and flexibility) for industrial networks, the MODBUS/TCP protocol allows for remote administration and control of devices over an Internet connection. It is important to note that not all Internet protocols are implemented in the module, for example HTTP and SMTP protocols are not available. Nevertheless, the efficiency, scalability and low cost of a MODBUS/TCP network make this an ideal solution for industrial applications.

The MVI46-MNETC module acts as an input/output module between devices on a MODBUS/TCP network and the Rockwell Automation backplane. The module uses an internal database to pass data and commands between the processor and the server devices on the MODBUS/TCP network.

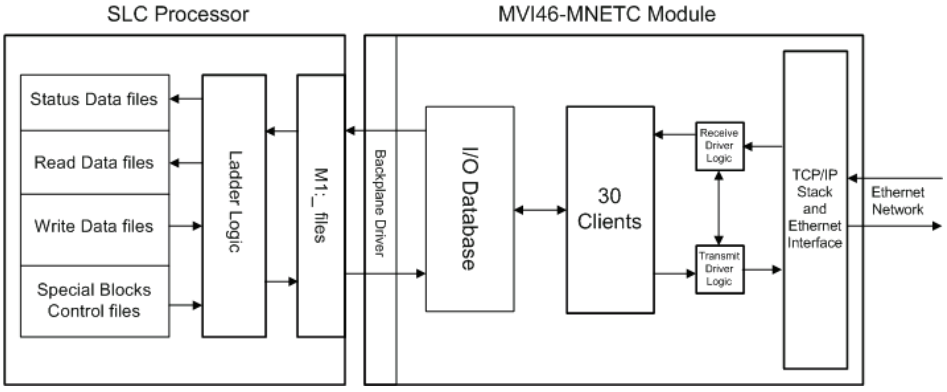
5.2.2 Backplane Data Transfer

The MVI46-MNETC module communicates directly over the SLC backplane. Data is paged between the module and the SLC processor across the backplane using the module's input and output images. Typical updates are in the range of 1 to 10 milliseconds.

This bi-directional transference of data is accomplished by the module filling in data in the module's input image to send to the processor. Data in the input image is placed in the M1 file in the processor by the ladder logic.

The processor inserts data to the module's output image to transfer to the module. The module's program extracts the data and places it in the module's internal database.

The following illustration shows the data transfer method used to move data between the SLC processor, the MVI46-MNETC module and the Modbus TCP/IP Network.



All data transferred between the module and the processor over the backplane is through the M1 file. Ladder logic in the SLC must interface the input and output image data with data defined in the SLC data files. All data used by the module is stored in its internal database. This database is defined as a virtual Modbus data table with addresses from 0 (40001 Modbus) to 4999 (45000 Modbus).

The following illustration shows the layout of the database:

Module's Internal Database Structure

Data contained in this database is paged through the input and output images by coordination of the SLC ladder logic and the MVI46-MNETC module's program. Up to 128 words of data can be transferred between the module and the processor per ladder logic scan. Each image has a defined structure depending on the data content and the function of the data transfer. The following table lists the block numbers used by the module.

Block Range	Descriptions
1000 to 1001	Output Initialization Blocks
2000 to 2029	Controls Modbus Commands as configured in the ladder logic.
5001 to 5016	Controls the Modbus Commands as listed in configuration file.
6000	Controls the selection of individual commands in the list, through set bits.
9990	Set new module IP address
9991	Get module IP address
9998	Warm-boot control block
9999	Cold-boot control block

The Block numbers represent data codes that the Module will recognize as instruction codes for performing a specific function.

The following topics describe these groups of blocks in more detail.

Normal Data Transfer

Normal data transfer includes the paging of the user data found in the module's internal database in registers 0 to 4999, and the status data. These data are transferred through reading and writing of the M1 files. No special block number is required to perform the normal data transfer.

The following topics describe the structure and function of each block.

Initialize Output Data

When the module performs a restart operation, it will request blocks of output data from the processor to initialize the module's output data. Use the **Initialize Output Data** parameter in the configuration file to bring the module to a known state after a restart operation. The following table describes the structure of the block used to request the data.

Offset	Description	Length
5000	Value 1000. The module is requesting initialization data.	1

The Ladder logic in the processor must recognize these blocks and place the correct information in the output image to return to the module. The following table describes the format of the returned write block.

Offset	Description	Length
5000	Value 1001. The SLC is responding with initialization data.	1

5.2.3 Command Control Blocks

Command Control Blocks are special optional blocks used to request special tasks from the module. The current version of the software supports the following command control blocks:

- Event Command Block
- Command Control
- Command Control Bits
- Set new module IP address
- Get module IP address
- Warm-boot
- Cold-boot

Event Command

Use blocks 2000 to 2029 to modify the Modbus Command parameters in the SLC's register files dynamically without the need to modify and reload the Configuration file MNET.CFG.

Event Request description (Write Block)

Event command control blocks send Modbus TCP/IP commands directly from the ladder logic to one of the clients on the module. The following table describes the format for these blocks.

Offset	Description	Length
5000	2000 to 2029 (last digits indicate which client to utilize)	1
5001 to 5004	IP Address	4
5005	Service Port	1
5006	Slave Address	1
5007	Internal DB Address	1
5008	Point Count	1
5009	Swap Code	1
5010	Modbus Function Code	1
5011	Device Database Address	1

Use the parameters passed with the block to construct the command. The **IP Address** for the node to reach on the network is entered in four registers (1 to 4). Each digit of the IP address is entered in the appropriate register.

For example, to interface with node 192.168.0.100, enter the values 192, 168, 0 and 100 in registers 1 to 4.

- The **Service Port** field selects the TCP service port on the server to connect. If the parameter is set to 502, a standard MBAP (Modbus API for network communications) message will be generated. All other service port values will generate a Modbus command message encapsulated in a TCP/IP packet.
- The **Internal DB Address** parameter specifies the module's database location containing the data to transfer.
- The **Point Count** parameter defines the quantity of points or registers to transfer.
- The **Swap Code** is used with Modbus functions 3 and 4 requests to change the word or byte order.
- The **Modbus Function Code** has one of the following values 1, 2, 3, 4, 5, 6, 15 or 16.
- The **Device Database Address** is the Modbus register or point in the remote server device containing the data to transfer.

Event Response description (Read Block)

When the module receives the block, it will process it and place it in the command queue. The third word of the block can be used by the ladder logic to determine if the command was added to the command queue of the module. The command will only fail if the command queue for the client is full (16commands for each queue).

The following table describes the format for this block.

Word	Description
5000	0 when Block execution is completed
5001	2000 to 2029. The last two digits indicate which Client was used.
5002	0 = Event Request Failed. 1 = Event Request Succeeded.

Modbus Command Control

Blocks 5001 to 5016 enable the user to list as many Modbus Commands in the configuration file MNET.CFG as desired (maximum quantity = 16 per client), then dynamically set the command execution quantity per client.

Command Request description (Write Block)

Command control blocks place commands in the command list into the command queue. The client has a command queue of up to 16 commands. The module services commands in the queue before the user defined command list. This gives high priority to commands in the queue. Commands placed in the queue through this mechanism must be defined in the module's command list.

Note: The Enable parameter in the configuration file's Modbus Command list must be set to either 0 or 1 and no other value. It does not matter if the value is set to 0 or 1 because the execution of the command is controlled by Blocks 5001 to 5016 and by Block 6000 in ladder logic.

With this Command Control Block, the commands may be placed in the command queue with an Enable parameter set to zero. One to 16 commands can be placed in the command queue with a single request. The following table describes the format for this block.

Offset	Description	Length
5000	5001 to 5016	1
5001	Client index to utilize [00 to 29]	1
5002	Command index [00]	1
5003	Command index [01]	1
5004	Command index [02]	1
5005	Command index [03]	1
5006	Command index [04]	1
5007	Command index [05]	1
5008	Command index [06]	1
5009	Command index [07]	1
5010	Command index [08]	1
5011	Command index [09]	1
5012	Command index [10]	1
5013	Command index [11]	1
5014	Command index [12]	1
5015	Command index [13]	1
5016	Command index [14]	1
5017	Command index [15]	1

The last digit in the block code defines the number of commands to process in the block. For example, a block code of 5003 contains 3 command indexes that are to be placed in the command queue. The Command index parameters in the block have a range of 0 to 15 and correspond to the module's command list entries.

Command Response description (Read Block)

The module responds to a command control block with a block containing the number of commands added to the command queue for the port. The following table describes the format for this block.

Offset	Description	Length
5000	0 when Block execution is completed	1
5001	5001 to 5016. The last two digits indicate which Modbus Commands were executed.	1
5002	0 = Event Request Failed. 1 = Event Request Succeeded.	1

Command Control Bits

Use block 6000 to list up to 16 Modbus Commands in the configuration file MNET.CFG, then dynamically set the specific commands to execute per client. The Modbus command execution will be continuous and sequential by the Client order and within each Client by the Command sequence order as listed in the configuration file. To dynamically modify the Client number and/or the Modbus Commands to execute from the list, modify the bit structure in the Client assigned registers and then re-activate the Block 6000.

Command Control Bits Request description (Write Block)

The Command Control Bits block places commands from the command list into the command queue. The client has a command queue of up to 16 commands. The module services commands in the queue before the user defined command list. This gives high priority to commands in the queue. Commands placed in the queue through this mechanism must be defined in the Modbus Command List section of the configuration file.

Note: The Enable parameter in the Modbus Command List section of the configuration file must be set to either 0 or 1 and no other value, otherwise the configuration file will not be properly read when the file is transferred to the module. The value can be either 0 or 1 because the execution of the command is controlled by Blocks 5001 to 5016 and by Block 6000.

The following table describes the format for this block.

Offset	Description	Length
5000	6000	1
5001 to 5030	Command Control Bits for clients [00 to 29]	30

Words 5001 to 5030 each contain set bits that correspond to the Command sequence position for each client in the configuration file.

For example, when Client 17 has in its configuration list all 16 Modbus commands and it is desired to execute only commands indexed 1, 8, 10 and 13, then word offset 5018 will have the following bits set:(MSB 0010 0101 0000 0010 LSB).

Command Control Bits Response description (Read Block)

The module responds to a command control block with a block containing the number of commands added to the command queue for the port. The following table describes the format for this block.

Offset	Description	Length
5000	0 when Block execution is completed	1
5001	6000 when Block execution is activated.	1
5002 to 5031	Command Control Bits for clients [00 to 29] that have been executed.	30

Set new module IP address

IP Set Request (Write Block)

Offset	Description	Length
5000	9990	1
5001	First digit of dotted IP address	1
5002	Second digit of dotted IP address	1
5003	Third digit of dotted IP address	1
5004	Last digit of dotted IP address	1

IP Set Response (Read Block)

Offset	Description	Length
5000	0 when completed.	1
5001	Write Block ID [9990]	1
5002	First digit of dotted IP address	1
5003	Second digit of dotted IP address	1
5004	Third digit of dotted IP address	1
5005	Last digit of dotted IP address	1

Get module IP address

IP Get Request (Write Block)

Offset	Description	Length
5000	9991	1

IP Get Response (Read Block)

Offset	Description	Length
5000	0 when complete.	1
5001	Write Block ID [9991]	1
5002	First digit of dotted IP address	1
5003	Second digit of dotted IP address	1
5004	Third digit of dotted IP address	1
5005	Last digit of dotted IP address	1

Warm Boot

This block is sent from the SLC processor to the module when the module is required to perform a warm-boot operation. This block is commonly sent to the module whenever data modifications are made in the data file register values. This will force the module to read the new data file values and to restart. The following table describes the structure of the control block.

Offset	Description	Length
5000	9998	1

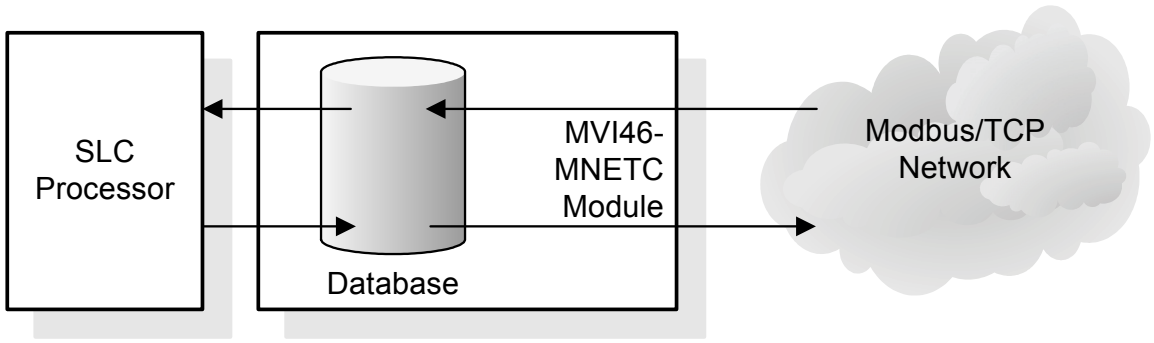
Cold Boot

This block is sent from the SLC processor to the module when the module is required to perform the cold boot (hardware reset) operation. This block is sent to the module when a hardware problem is detected by the ladder logic that requires a hardware reset. The following table describes the structure of the control block.

Offset	Description	Length
5000	9999	1

5.2.4 Data Flow between MVI46-MNETC Module and SLC Processor

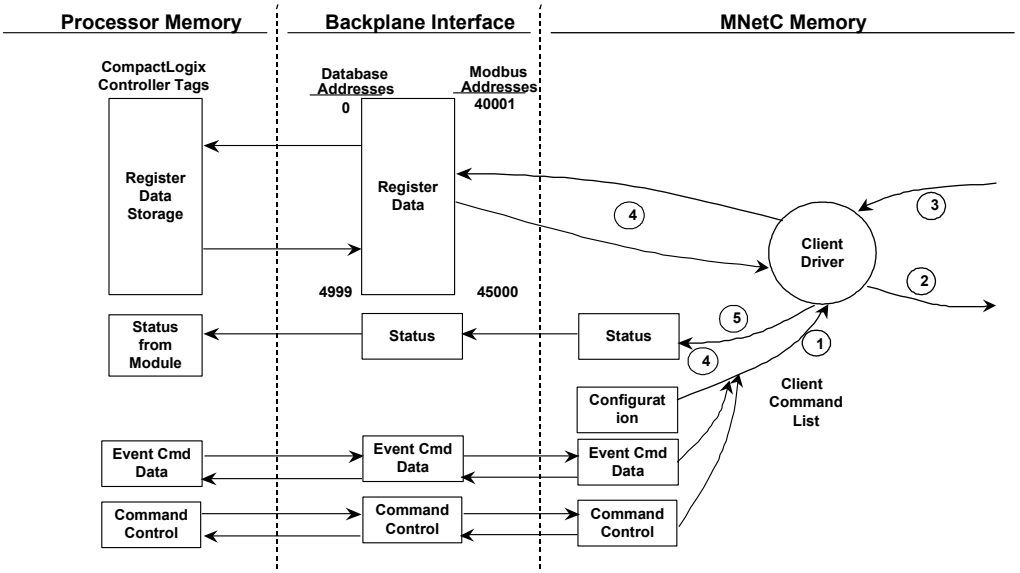
The following topics describe the flow of data between the two pieces of hardware (SLC processor and MVI46-MNETC module) and other nodes on the Modbus TCP/IP network. The module contains up to 30 clients, which can generate either MBAP (Modbus API for network communications) or MNET requests dependent on the service port selected in the command.



The following topics discuss the operation of the client drivers.

Client Driver

In the client driver, the MVI46-MNETC module issues read or write commands to servers on the Modbus TCP/IP network using 30 simulated clients. These commands are user configured in the module via the Client Command List for each client received from the module's configuration file (MNET.CFG) or issued directly from the SLC processor (event command control). Command status is returned to the processor for each individual command in the command list status block. The location of this status block in the module's internal database is user defined in the module's configuration file. The following flow chart and associated table describe the flow of data into and out of the module.



Step	Description
1	The client driver obtains configuration data from the MNET.CFG file when the module restarts. The configuration data obtained includes the timeout parameters and the Modbus Command List. These values are used by the driver to determine the type of commands to issue to the other nodes on the Modbus TCP/IP network.
2	When configured, the client driver begins transmitting read and/or write commands to the other nodes on the network. If writing data to another node, the data for the write command is obtained from the module's internal database to build the command.
3	Presuming successful processing by the node specified in the command, a response message is received into the client driver for processing.
4	Data received from the node on the network is passed into the module's internal database, assuming a read command.
5	Status data is returned to the SLC processor for the client and a Command List error table can be established in the module's internal database.

Client Command List

In order for the client to function, the module's Client Command List must be defined. This list contains up to 16 individual entries, with each entry containing the information required to construct a valid command. This includes the following:

- Command enable mode ((0) disabled, (1) continuous [controlled by SLC Data files exclusively]).
- IP address and service port to connect to on the remote server
- Slave Node Address
- Command Type - Read or Write up to 125 words per command
- Database Source and Destination Register Address - Determines where data will be placed and/or obtained
- Count - Select the number of words to be transferred - 1 to 125
- Poll Delay - 1/10th seconds

Client Command Errors

You can use the Client Command Error Pointer, which is configured for each client in the MNET.CFG file. This pointer references the offset register where all command error codes will be stored. This means that the first register refers to command 1 and so on.

Offset	Description
4110	Client 0 Command Error
4140	Client 1 Command Error
4170	Client 2 Command Error
…	….
…	…

For every command that has an error, the module automatically sets the poll delay parameter to 30 seconds. This instructs the module to wait 30 seconds until it attempts to issue the command again.

As the list is read in from the configuration file and as the commands are processed, an error value is maintained in the module for each command. This error list can be transferred to the processor. The errors generated by the module are displayed in the following table.

Standard Modbus Protocol Errors

Code	Description
1	Illegal Function
2	Illegal Data Address
3	Illegal Data Value
4	Failure in Associated Device
5	Acknowledge
6	Busy, Rejected Message

Module Communication Error Codes

Code	Description
-1	CTS modem control line not set before transmit
-2	Timeout while transmitting message
-11	Timeout waiting for response after request
253	Incorrect slave address in response
254	Incorrect function code in response
255	Invalid CRC/LRC value in response

MNET Client Specific Errors

Code	Description
-33	Failed to connect to server specified in command
-36	MNET command response timeout
-37	TCP/IP connection ended before session finished

Command List Entry Errors

Code	Description
-40	Too few parameters
-41	Invalid enable code
-42	Internal address > maximum address
-43	Invalid node address (<0 or >255)
-44	Count parameter set to 0
-45	Invalid function code
-46	Invalid swap code
-47	ARP could not resolve MAC from IP (bad IP address, not part of a network, invalid parameter to ARP routine).
-48	Error during ARP operation: the response to the ARP request did not arrive to the module after a 5 second timeout.

Note: When the client gets error 47 or 48, it places the command offline for 30 seconds so it will not keep waiting for servers that are not present on the network, blocking the commands being executed to valid/live servers on the network.

5.3 Cable Connections

The MVI46-MNETC module has the following communication connections on the module:

- One Ethernet port (RJ45 connector)
- One RS-232 Configuration/Debug port (RJ45 connector)

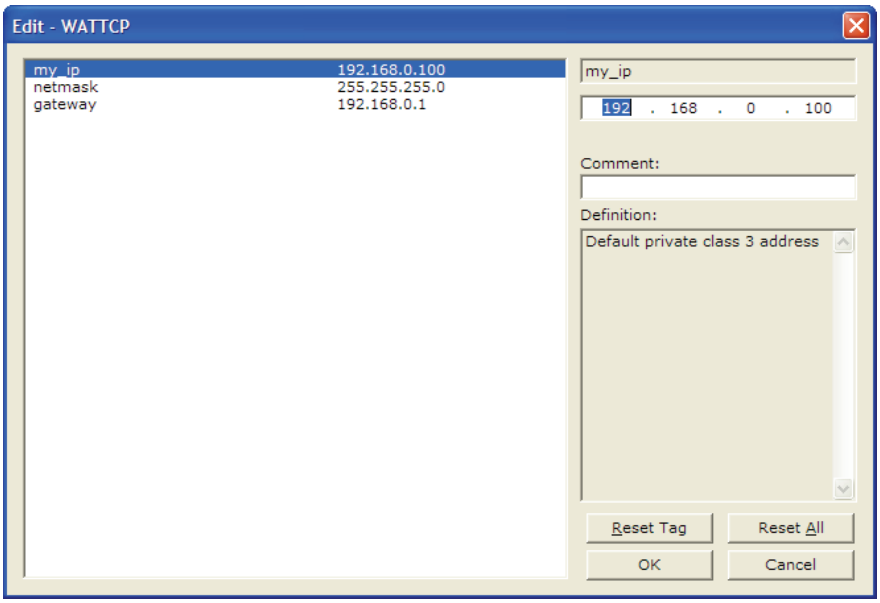
5.3.1 Ethernet Connection

The MVI46-MNETC module has an RJ45 port located on the front of the module labeled "Ethernet", for use with the TCP/IP network. The module is connected to the Ethernet network using an Ethernet cable between the module's Ethernet port and an Ethernet switch or hub.

Note: Depending on hardware configuration, you may see more than one RJ45 port on the module. The Ethernet port is labeled "Ethernet".
Warning: The MVI46-MNETC module is NOT compatible with Power Over Ethernet (IEEE802.3af / IEEE802.3at) networks. Do NOT connect the module to Ethernet devices, hubs, switches or networks that supply AC or DC power over the Ethernet cable. Failure to observe this precaution may result in damage to hardware, or injury to personnel.
Important: The module requires a static (fixed) IP address that is not shared with any other device on the Ethernet network. Obtain a list of suitable IP addresses from your network administrator BEFORE configuring the Ethernet port on this module.

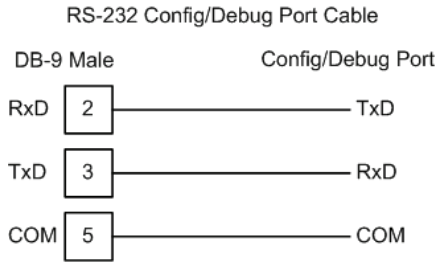
Ethernet Port Configuration - wattcp.cfg

The wattcp.cfg file must be set up properly in order to use a TCP/IP network connection. You can view the current network configuration using an ASCII terminal by selecting **[@]** (Network Menu) and **[V]** (View) options when connected to the Debug port.



5.3.2 RS-232 Configuration/Debug Port

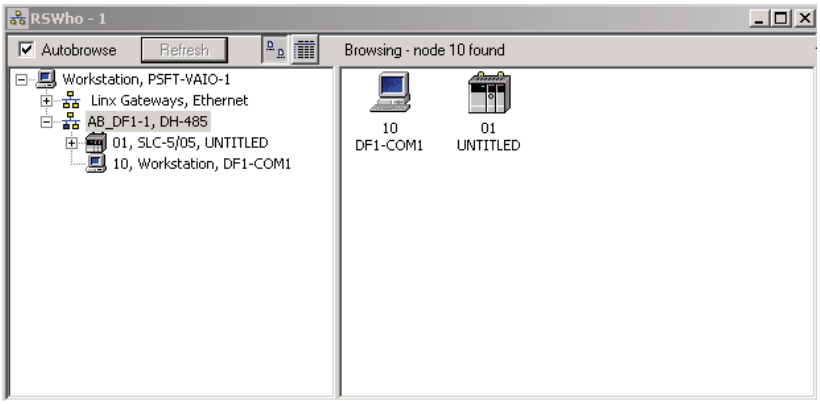
This port is physically an RJ45 connection. An RJ45 to DB-9 adapter cable is included with the module. This port permits a PC based terminal emulation program to view configuration and status data in the module and to control the module. The cable for communications on this port is shown in the following diagram:



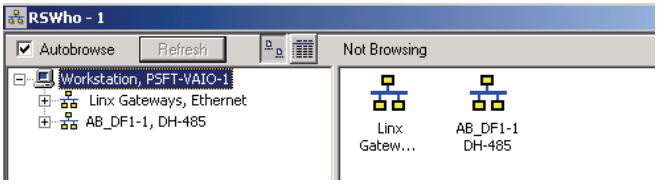
Disabling the RSLinx Driver for the Com Port on the PC

The communication port driver in RSLinx can occasionally prevent other applications from using the PC's COM port. If you are not able to connect to the module's configuration/debug port using ProSoft Configuration Builder (PCB), HyperTerminal or another terminal emulator, follow these steps to disable the RSLinx Driver.

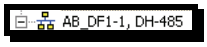
1. Open RSLinx and go to Communications>RSWho
2. Make sure that you are not actively browsing using the driver that you wish to stop. The following shows an actively browsed network:



3 Notice how the DF1 driver is opened, and the driver is looking for a processor on node 1. If the network is being browsed, then you will not be able to stop this driver. To stop the driver your RSWho screen should look like this:

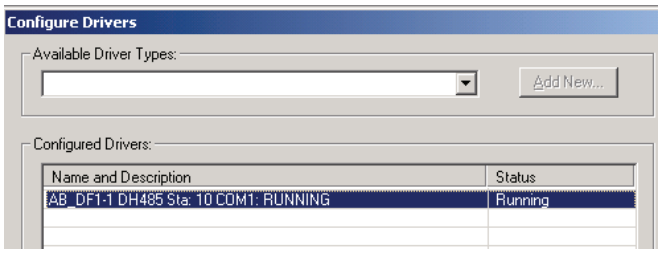


Branches are displayed or hidden by clicking on the ⊞ or the ⊟ icons.



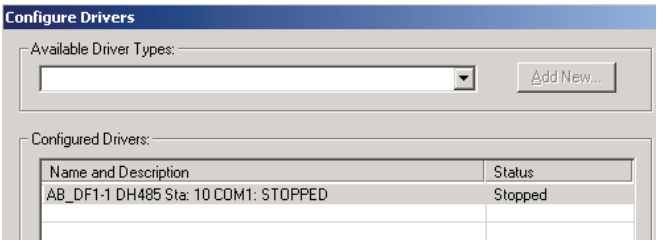
4 When you have verified that the driver is not being browsed, go to

Communications>Configure Drivers
You may see something like this:



If you see the status as running, you will not be able to use this com port for anything other than communication to the processor. To stop the driver press the "Stop" on the side of the window:

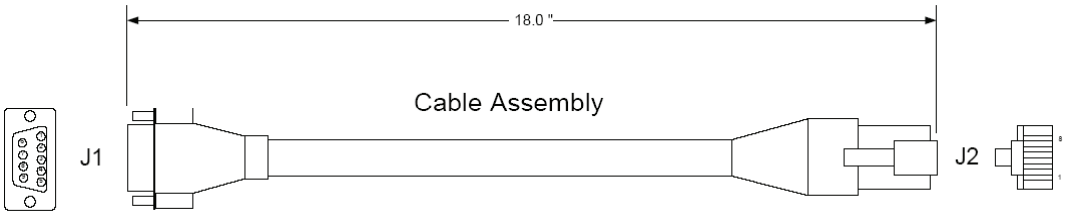
5 After you have stopped the driver you will see the following:

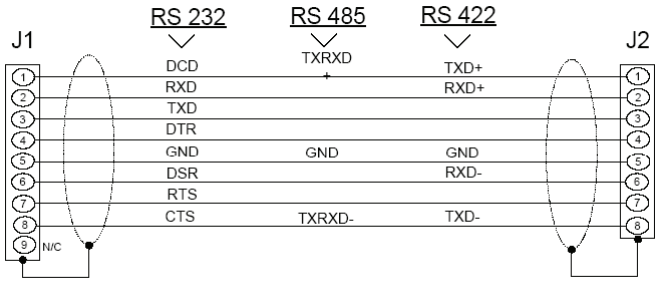


6 Upon seeing this, you may now use that com port to connect to the debug port of the module.

Note: You may need to shut down and restart your PC before it will allow you to stop the driver (usually only on Windows NT machines). If you have followed all of the above steps, and it will not stop the driver, then make sure you do not have RSLogix open. If RSLogix is not open, and you still cannot stop the driver, then reboot your PC.

5.3.3 DB9 to RJ45 Adaptor (Cable 14)





Wiring Diagram

6 Support, Service & Warranty

In This Chapter

- How to Contact Us: Technical Support 77
- Return Material Authorization (RMA) Policies and Conditions 78
- LIMITED WARRANTY 79

ProSoft Technology, Inc. (ProSoft) is committed to providing the most efficient and effective support possible. Before calling, please gather the following information to assist in expediting this process:

1 Product Version Number
2 System architecture
3 Network details

If the issue is hardware related, we will also need information regarding:

1 Module configuration and contents of file
   - Module Operation
   - Configuration/Debug status information
   - LED patterns

2 Information about the processor and user data files as viewed through and LED patterns on the processor.
3 Details about the serial devices interfaced, if any.

6.1 How to Contact Us: Technical Support

Internet	Web Site: www.prosoft-technology.com/support
	E-mail address: support@prosoft-technology.com

Asia Pacific

+603.7724.2080, support.asia@prosoft-technology.com
Languages spoken include: Chinese, English

Europe (location in Toulouse, France)

+33 (0) 5.34.36.87.20, support.EMEA@prosoft-technology.com
Languages spoken include: French, English

North America/Latin America (excluding Brasil) (location in California)

+1.661.716.5100, support@prosoft-technology.com
Languages spoken include: English, Spanish
For technical support calls within the United States, an after-hours answering system allows pager access to one of our qualified technical and/or application support engineers at any time to answer your questions.

Brasil (location in Sao Paulo)

+55-11-5084-5178, eduardo@prosoft-technology.com
Languages spoken include: Portuguese, English

6.2 Return Material Authorization (RMA) Policies and Conditions

The following RMA Policies and Conditions (collectively, "RMA Policies") apply to any returned Product. These RMA Policies are subject to change by ProSoft without notice. For warranty information, see "Limited Warranty". In the event of any inconsistency between the RMA Policies and the Warranty, the Warranty shall govern.

6.2.1 All Product Returns:

a) In order to return a Product for repair, exchange or otherwise, the Customer must obtain a Returned Material Authorization (RMA) number from ProSoft and comply with ProSoft shipping instructions.
b) In the event that the Customer experiences a problem with the Product for any reason, Customer should contact ProSoft Technical Support at one of the telephone numbers listed above (page 77). A Technical Support Engineer will request that you perform several tests in an attempt to isolate the problem. If after completing these tests, the Product is found to be the source of the problem, we will issue an RMA.
c) All returned Products must be shipped freight prepaid, in the original shipping container or equivalent, to the location specified by ProSoft, and be accompanied by proof of purchase and receipt date. The RMA number is to be prominently marked on the outside of the shipping box. Customer agrees to insure the Product or assume the risk of loss or damage in transit. Products shipped to ProSoft using a shipment method other than that specified by ProSoft or shipped without an RMA number will be returned to the Customer, freight collect. Contact ProSoft Technical Support for further information.
d) A 10% restocking fee applies to all warranty credit returns whereby a Customer has an application change, ordered too many, does not need, and so on.

6.2.2 Procedures for Return of Units Under Warranty:

A Technical Support Engineer must approve the return of Product under ProSoft's Warranty:

a) A replacement module will be shipped and invoiced. A purchase order will be required.
b) Credit for a product under warranty will be issued upon receipt of authorized product by ProSoft at designated location referenced on the Return Material Authorization.

6.2.3 Procedures for Return of Units Out of Warranty:

a) Customer sends unit in for evaluation
b) If no defect is found, Customer will be charged the equivalent of $100 USD, plus freight charges, duties and taxes as applicable. A new purchase order will be required.

c) If unit is repaired, charge to Customer will be 30% of current list price (USD) plus freight charges, duties and taxes as applicable. A new purchase order will be required or authorization to use the purchase order submitted for evaluation fee.

The following is a list of non-repairable units:

- 3150 - All
- 3750
- 3600 - All
- 3700
- 3170 - All
- 3250
- 1560 - Can be repaired, only if defect is the power supply
- 1550 - Can be repaired, only if defect is the power supply
- 3350
- 3300
- 1500 - All

6.3 LIMITED WARRANTY

This Limited Warranty ("Warranty") governs all sales of hardware, software and other products (collectively, "Product") manufactured and/or offered for sale by ProSoft, and all related services provided by ProSoft, including maintenance, repair, warranty exchange, and service programs (collectively, "Services"). By purchasing or using the Product or Services, the individual or entity purchasing or using the Product or Services ("Customer") agrees to all of the terms and provisions (collectively, the "Terms") of this Limited Warranty. All sales of software or other intellectual property are, in addition, subject to any license agreement accompanying such software or other intellectual property.

6.3.1 What Is Covered By This Warranty

a) *Warranty On New Products*: ProSoft warrants, to the original purchaser, that the Product that is the subject of the sale will (1) conform to and perform in accordance with published specifications prepared, approved and issued by ProSoft, and (2) will be free from defects in material or workmanship; provided these warranties only cover Product that is sold as new. This Warranty expires three years from the date of shipment (the "Warranty Period"). If the Customer discovers within the Warranty Period a failure of the Product to conform to specifications, or a defect in material or workmanship of the Product, the Customer must promptly notify ProSoft by fax, email or telephone. In no event may that notification be received by ProSoft later than 39 months. Within a reasonable time after notification, ProSoft will correct any failure of the Product to conform to specifications or any defect in material or workmanship of the Product, with either new or used replacement parts. Such repair, including both parts and labor, will be performed at ProSoft's expense. All warranty service will be performed at service centers designated by ProSoft.

b) *Warranty On Services*: Materials and labor performed by ProSoft to repair a verified malfunction or defect are warranteed in the terms specified above for new Product, provided said warranty will be for the period remaining on the original new equipment warranty or, if the original warranty is no longer in effect, for a period of 90 days from the date of repair.

6.3.2 What Is Not Covered By This Warranty

a) ProSoft makes no representation or warranty, expressed or implied, that the operation of software purchased from ProSoft will be uninterrupted or error free or that the functions contained in the software will meet or satisfy the purchaser's intended use or requirements; the Customer assumes complete responsibility for decisions made or actions taken based on information obtained using ProSoft software.
b) This Warranty does not cover the failure of the Product to perform specified functions, or any other non-conformance, defects, losses or damages caused by or attributable to any of the following: (i) shipping; (ii) improper installation or other failure of Customer to adhere to ProSoft's specifications or instructions; (iii) unauthorized repair or maintenance; (iv) attachments, equipment, options, parts, software, or user-created programming (including, but not limited to, programs developed with any IEC 61131-3, "C" or any variant of "C" programming languages) not furnished by ProSoft; (v) use of the Product for purposes other than those for which it was designed; (vi) any other abuse, misapplication, neglect or misuse by the Customer; (vii) accident, improper testing or causes external to the Product such as, but not limited to, exposure to extremes of temperature or humidity, power failure or power surges; or (viii) disasters such as fire, flood, earthquake, wind and lightning.
c) The information in this Agreement is subject to change without notice. ProSoft shall not be liable for technical or editorial errors or omissions made herein; nor for incidental or consequential damages resulting from the furnishing, performance or use of this material. The user guide included with your original product purchase from ProSoft contains information protected by copyright. No part of the guide may be duplicated or reproduced in any form without prior written consent from ProSoft.

6.3.3 Disclaimer Regarding High Risk Activities

Product manufactured or supplied by ProSoft is not fault tolerant and is not designed, manufactured or intended for use in hazardous environments requiring fail-safe performance including and without limitation: the operation of nuclear facilities, aircraft navigation of communication systems, air traffic control, direct life support machines or weapons systems in which the failure of the product could lead directly or indirectly to death, personal injury or severe physical or environmental damage (collectively, "high risk activities"). ProSoft specifically disclaims any express or implied warranty of fitness for high risk activities.

6.3.4 Intellectual Property Indemnity

Buyer shall indemnify and hold harmless ProSoft and its employees from and against all liabilities, losses, claims, costs and expenses (including attorney's fees and expenses) related to any claim, investigation, litigation or proceeding (whether or not ProSoft is a party) which arises or is alleged to arise from Buyer's acts or omissions under these Terms or in any way with respect to the Products. Without limiting the foregoing, Buyer (at its own expense) shall indemnify and hold harmless ProSoft and defend or settle any action brought against such Companies to the extent based on a claim that any Product made to Buyer specifications infringed intellectual property rights of another party. ProSoft makes no warranty that the product is or will be delivered free of any person's claiming of patent, trademark, or similar infringement. The Buyer assumes all risks (including the risk of suit) that the product or any use of the product will infringe existing or subsequently issued patents, trademarks, or copyrights.

a) Any documentation included with Product purchased from ProSoft is protected by copyright and may not be duplicated or reproduced in any form without prior written consent from ProSoft.
b) ProSoft's technical specifications and documentation that are included with the Product are subject to editing and modification without notice.
c) Transfer of title shall not operate to convey to Customer any right to make, or have made, any Product supplied by ProSoft.
d) Customer is granted no right or license to use any software or other intellectual property in any manner or for any purpose not expressly permitted by any license agreement accompanying such software or other intellectual property.
e) Customer agrees that it shall not, and shall not authorize others to, copy software provided by ProSoft (except as expressly permitted in any license agreement accompanying such software); transfer software to a third party separately from the Product; modify, alter, translate, decode, decompile, disassemble, reverse-engineer or otherwise attempt to derive the source code of the software or create derivative works based on the software; export the software or underlying technology in contravention of applicable US and international export laws and regulations; or use the software other than as authorized in connection with use of Product.
f) **Additional Restrictions Relating To Software And Other Intellectual Property**

   In addition to compliance with the Terms of this Warranty, Customers purchasing software or other intellectual property shall comply with any license agreement accompanying such software or other intellectual property. Failure to do so may void this Warranty with respect to such software and/or other intellectual property.

6.3.5 Disclaimer of all Other Warranties

The Warranty set forth in What Is Covered By This Warranty (page 79) are in lieu of all other warranties, express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

6.3.6 Limitation of Remedies **

In no event will ProSoft or its Dealer be liable for any special, incidental or consequential damages based on breach of warranty, breach of contract, negligence, strict tort or any other legal theory. Damages that ProSoft or its Dealer will not be responsible for included, but are not limited to: Loss of profits; loss of savings or revenue; loss of use of the product or any associated equipment; loss of data; cost of capital; cost of any substitute equipment, facilities, or services; downtime; the claims of third parties including, customers of the Purchaser; and, injury to property.

** Some areas do not allow time limitations on an implied warranty, or allow the exclusion or limitation of incidental or consequential damages. In such areas, the above limitations may not apply. This Warranty gives you specific legal rights, and you may also have other rights which vary from place to place.

6.3.7 Time Limit for Bringing Suit

Any action for breach of warranty must be commenced within 39 months following shipment of the Product.

6.3.8 No Other Warranties

Unless modified in writing and signed by both parties, this Warranty is understood to be the complete and exclusive agreement between the parties, suspending all oral or written prior agreements and all other communications between the parties relating to the subject matter of this Warranty, including statements made by salesperson. No employee of ProSoft or any other party is authorized to make any warranty in addition to those made in this Warranty. The Customer is warned, therefore, to check this Warranty carefully to see that it correctly reflects those terms that are important to the Customer.

6.3.9 Allocation of Risks

This Warranty allocates the risk of product failure between ProSoft and the Customer. This allocation is recognized by both parties and is reflected in the price of the goods. The Customer acknowledges that it has read this Warranty, understands it, and is bound by its Terms.

6.3.10 Controlling Law and Severability

This Warranty shall be governed by and construed in accordance with the laws of the United States and the domestic laws of the State of California, without reference to its conflicts of law provisions. If for any reason a court of competent jurisdiction finds any provisions of this Warranty, or a portion thereof, to be unenforceable, that provision shall be enforced to the maximum extent permissible and the remainder of this Warranty shall remain in full force and effect. Any cause of action with respect to the Product or Services must be instituted in a court of competent jurisdiction in the State of California.

Index

[

[MNET Client x Commands] • 27
[MNET Client x] • 25
[Module] • 23
[Static ARP Table] • 24

A

About the MODBUS/TCP Protocol • 61
Adding the Module to an Existing Project • 41
All Product Returns: • 78
Allocation of Risks • 82
ARP Timeout • 26

B

Backplane Data Transfer • 62
Battery Life Advisory • 3

C

Cable Connections • 73
Client Driver • 70
Cold Boot • 69
Command Control Bits • 67
Command Control Blocks • 64
Command Entry Formats • 28
Command Error Delay • 27
Command Error Pointer • 25
Command List Menu • 53
Command List Overview • 27
Commands Supported by the Module • 28
Comment • 32
Configuring Module Parameters • 22
Configuring the Floating Point Data Transfer • 34
Configuring the MVI46-MNETC Module • 19
Configuring the RSLinx Driver for the PC COM Port • 15
Connect your PC to the Module • 17
Connect your PC to the Processor • 13
Controlling Law and Severability • 82

D

Data Flow between MVI46-MNETC Module and SLC Processor • 69
DB9 to RJ45 Adaptor (Cable 14) • 76
Diagnostics and Troubleshooting • 7, 45
Disabling the RSLinx Driver for the Com Port on the PC • 74
Disclaimer of all Other Warranties • 81
Disclaimer Regarding High Risk Activities • 80
Download the Project to the Module • 39
Download the Sample Program to the Processor • 14
Duplex/Speed Code • 24

E

Enable • 29
Error/Status Pointer • 23, 25
Ethernet Configuration • 33
Ethernet Connection • 73
Ethernet LED Indicators • 58
Ethernet Port Configuration - wattcp.cfg • 73
Event Command • 64
Exiting the Program • 51

F

Failure Flag Count • 23
Float Flag • 26
Float Offset • 26
Float Start • 26
Functional Overview • 7, 61
Functional Specifications - MVI-MNETC • 60

G

General Concepts • 61
General Specifications • 59
General Specifications - Modbus TCP/IP Client • 59
Get module IP address • 68
Guide to the MVI46-MNETC User Manual • 7

H

Hardware MAC Address • 25
Hardware Specifications • 60
How to Contact Us
    Technical Support • 77, 78

I

Important Installation Instructions • 2
Initialize Output Data • 24, 64
Install ProSoft Configuration Builder Software • 10
Install the Module in the Rack • 12
Intellectual Property Indemnity • 81
Internal Address • 29
IP Address • 24

K

Keystrokes • 48

L

Ladder Logic • 41
LED Status Indicators • 7, 56
Limitation of Remedies ** • 82
LIMITED WARRANTY • 79

M

M1 Write Size • 23
Main Menu • 49
Master Command Error List Menu • 54
MB Address in Device • 32
MBAP Port Override • 27

Minimum Command Delay • 25
Modbus Command Control • 65
Modbus Database View • 52
Modbus Function • 31
Moving Back Through 5 Pages of Registers • 52
Moving Forward Through 5 Pages of Registers • 52
MVI (Multi Vendor Interface) Modules • 2

N

Navigation • 48
Network Menu • 55
No Other Warranties • 82
Node IP Address • 31
Normal Data Transfer • 63

O

Opening the Command Error List Menu • 50
Opening the Command List Menu • 50
Opening the Database Menu • 50

P

Package Contents • 10
Pinouts • 73, 76
Poll Interval • 30
Printing a Configuration File • 23
Procedures for Return of Units Out of Warranty: • 78
Procedures for Return of Units Under Warranty: • 78
Product Specifications • 7, 59
ProSoft Technology® Product Documentation • 4

R

Reading Status Data from the Module • 45
Receiving the Configuration File • 50
Redisplaying the Current Page • 52
Redisplaying the Menu • 53, 54
Reference • 7, 59
Reg Count • 30
Required Hardware • 45
Resetting diagnostic data • 50
Response Timeout • 25
Retry Count • 25
Return Material Authorization (RMA) Policies and Conditions • 78
Returning to the Main Menu • 53, 54, 55
RS-232 Configuration/Debug Port • 74

S

Sending the Configuration File • 50
Service Port • 31
Set Module Parameters • 22
Set new module IP address • 68
Set Up the Project • 19
Setting Command Control Bits • 34
Setting Jumpers • 11
Slave Address • 31
Start Here • 7, 9
Support, Service & Warranty • 7, 77
Swap Code • 30
System Requirements - MVI46 PCB • 9

T

The Configuration/Debug Menu • 46
Time Limit for Bringing Suit • 82
Transferring WATTCP.CFG to the module • 55
Transferring WATTCP.CFG to the PC • 55

U

Using ProSoft Configuration Builder • 19
Using the Diagnostic Window in ProSoft Configuration Builder • 46

V

Viewing Block Transfer Statistics • 49
Viewing Client Configuration • 51
Viewing Client Status • 51
Viewing Data in ASCII (Text) Format • 53
Viewing Data in Decimal Format • 53
Viewing Data in Floating Point Format • 53
Viewing Data in Hexadecimal Format • 53
Viewing Module Configuration • 49
Viewing Register Pages • 52
Viewing the Next 100 Registers of Data • 53
Viewing the Next Page of Commands • 54
Viewing the Previous 100 Registers of Data • 52
Viewing the Previous Page of Commands • 54
Viewing the WATTCP.CFG file on the module • 55
Viewing Version Information • 50

W

Warm Boot • 68
Warm Booting the Module • 51
Warnings • 2
What Is Covered By This Warranty • 79, 81
What Is Not Covered By This Warranty • 80

Y

Your Feedback Please • 4